(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 059 433
A1

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 82101446.1

(22) Date of filing: 25.02.82

(51) Int. Cl.³: G 01 P 3/489

(30) Priority: 28.02.81 JP 29271/81

(43) Date of publication of application:
08.09.82 Bulletin 82/36

(84) Designated Contracting States:
CH DE LI SE

(71) Applicant: Hitachi, Ltd.
5-1, Marunouchi 1-chome
Chiyoda-ku Tokyo 100(JP)

(72) Inventor: Ohmae, Tsutomu
21-2, Motomiyacho-4-chome
Hitachi-shi(JP)

(72) Inventor: Matsuda, Toshihiko
10-2, Suehirocho-3-chome
Hitachi-shi(JP)

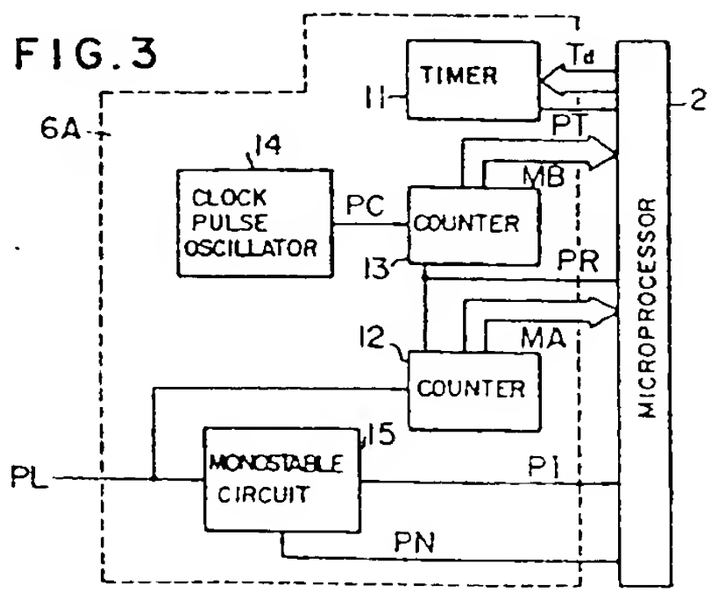
(72) Inventor: Tachikawa, Makoto
7-8, Suwacho-4-chome
Hitachi-shi(JP)

(74) Representative: Ebbinghaus, Dieter et al,
K.L. Schiff Dr. A. von Funer Dipl.-Ing. P. Strehl Dr. U. Schübel-Hopf Dipl.-Ing. D. Ebbinghaus Dr. Ing. D. Finck
Patentanwälte Mariahilfplatz 2 & 3
D-8000 München 90(DE)

(54) Speed detecting method and apparatus.

(57) A counter (12) is provided to count an output pulse which a pulse generator (5) generates each time a vehicle moves by a predetermined distance. A microprocessor (2) is provided to actuate a timer (11) in synchronism with the leading edges of the output pulse from the pulse generator (5). The timer (11) is set to a constant time interval during which the speed of the vehicle is to be detected. The microprocessor (2) calculates the speed of the vehicle from the count of the counter (12) which count is a value during the time interval from when the timer (11) starts to operate to when the pulse generator (5) generates a pulse just before or after the end point of the set time interval. Thus, the speed of the vehicle can be detected with good resolution and precision.

FIG.3



EP 0 059 433 A1

SPEED DETECTING METHOD AND APPARATUS

This invention relates to a speed detecting method and apparatus suitable for use in controlling the speed of a vehicle or a rotating body in a digital manner.

Generally, in order to detect the rotational speed of a motor as a digital signal, a pulse generator is used which generates a pulse of a frequency proportional to the speed of rotation. The pulse generator generates a single pulse each time the motor rotates by 1/n of one revolution (n is a large integer). To detect the speed of rotation by output pulse from a pulse generator, there is used a pulse number counting method or a pulse interval counting method.

The pulse number counting method is to count the number of output pulses which the pulse generator generates during a constant time, thereby detecting the speed of rotation. The pulse interval counting method is to count a clock pulse of a constant frequency during the interval between output pulses which the pulse generator generates, thereby detecting the speed of rotation.

However, both methods have the following drawbacks.

In the pulse number counting method, the number of pulses generated within a constant time at

a low speed is small and thus the resolution of detecting speed is poor. In order to increase the resolution at a low speed, the constant time is extended or the number of pulses the pulse generator generates at each revolution is increased. Extension of the constant time, however, will increase the time which is required for the speed signal to be obtained, and decrease the control response to the motor. Moreover, it is difficult to increase the number of pulses which the pulse generator generates at each revolution from the its construction viewpoint.

On the other hand, in the pulse interval counting method, the count of clock pulses becomes small at a high speed at which the interval between output pulses from the pulse generator is narrow. Therefore, the resolution of speed detection is poor.

The technique for obviating the drawbacks is described in the literature, U.S. Patent No. 3,210,123 "High Speed Frequency Computing Apparatus". In this literature, the frequency is measured from the number of periods of a sinusoidal signal which is counted during a set time interval. Specifically, counting of the number of periods of a sinusoidal wave signal is started at the measuring start point of the set time interval in synchronism with the sinusoidal wave signal, and ended at the end point of the set time interval, in which case if a fraction of the sinusoidal wave signal

occurs at the end point the count is compensated therefor by the ratio of the time corresponding to the fraction to the preceding period before measuring the frequency.

In the method described in the above literature, there is a problem that the precision of detection is reduced when the period at the end point of the set time interval and the period just therebefore are changed.

It is an object of this invention to provide a speed detecting method and apparatus which is capable of detecting the speed with good resolution and precision by counting the output pulse from the pulse generator.

A feature of this invention is that the speed is detected by counting the number of pulses which a pulse generator generates during a time interval from when the pulse generator generates a pulse at the measuring start point of a set time interval or just before or after the measuring start point, to when the pulse generator generates a pulse just before or after the end point of the set time interval.

Another feature of this invention is that the set time interval is changed in accordance with the speed of a vehicle.

Other objects and features of this invention will become apparent from the following description taken with the accompanying drawings in which:

Fig. 1 is a timing chart useful for explaining

the principle of this invention;

Fig. 2 shows an arrangement of one embodiment of this invention;

Fig. 3 shows an arrangement of one example of a speed detecting circuit according to this invention;

Figs. 4 to 6 are flowcharts useful for explaining the operation of the arrangement of Fig. 3;

Figs. 7 to 10 are timing charts useful for explaining the operation of Fig. 3;

Fig. 11 is an arrangement of another example of a speed detecting circuit according to this invention;

Figs. 12 and 13 are flowcharts useful for explaining the operation of the arrangement of Fig. 11;

Fig. 14 is a timing chart useful for explaining the operation of the arrangement of Fig. 11;

Fig. 15 is an arrangement of still another example of a speed detecting circuit according to this invention;

Figs. 16 and 17 are flowcharts useful for explaining the operation of the arrangement of Fig. 15;

Fig. 18 is a timing chart useful for explaining the operation of the arrangement of Fig. 15;

Figs. 19 and 20 are a flowchart and a timing chart useful for explaining another detecting method according to this invention;

Fig. 21 is an arrangement of further example of a speed detecting circuit according to this invention;

Figs. 22 and 23 are flow charts useful for explaining the operation of the arrangement of Fig. 21; and

Fig. 24 is a timing chart useful for explaining the operation of the arrangement of Fig. 21.

First, the fundamental idea of this invention will be described with reference to Fig. 1. Fig. 1 shows six methods A to F according to this invention. The method A thereof which is most easy to understand will be described.

In method A, the measurement of a set time interval Td and the counting of a pulse PL are started in synchronism with the leading edge of the output pulse PL from a pulse generator, and the counting of the pulse PL is stopped just after the end point of the set time interval Td in synchronism with the pulse PL which the pulse generator generates. The time during which the pulse PL is counted (speed detecting time) is $Td + \Delta T_1$. The count of the pulse PL within the time $Td + \Delta T_1$ is represented by M_1. A clock pulse CP is counted during the counting time $Td + \Delta T_1$, and the count, M_2 of the clock pulse CP is proportional to the counting time $Td + \Delta T_1$. In this invention, the speed detected value, Nf is determined from the following equation by substituting the values M_1 and M_2 thereinto:

$$Nf = k \frac{M_1}{M_2} \qquad \text{----------} \quad (1)$$

k: constant

Thus, the speed detecting time is the sum of the set time interval (a constant value) Td and compensation time ΔT_1. The maximum value of the compensation time ΔT_1 is substantially equal to the interval of the pulse PL. The compensation time ΔT_1 is the maximum at the lowest speed. However, if the count of pulse PL during the set time interval Td at the lowest speed is represented by M_{1L}, then $\Delta T \fallingdotseq Td/M_{1L}$, thus the ΔT being relatively small. Therefore, the speed can be detected with satisfactory control response upon the control of the vehicle.

On the other hand, as to the resolution, when the revolution rate is low, the count M_1 is small, but the rate of change of the compensation time ΔT_1 is large, and thus change of the count M_2 is great. Great change of the count M_2 means that the number of variation steps which M_1/M_2 can take is increased to increase the resolution.

As to the precision, the speed detecting time $Td + \Delta T_1$ is synchronized with the output pulse PL from the pulse generator, and the pulse PL duration is proportional to the distance by which the vehicle moves, the speed being detected from the accurate distance per $(Td + \Delta T_1)$, so that the precision of detection is high. In this case, the count of the pulse PL during the speed detecting time $Td + \Delta T_1$ takes three different values and even if error occurs in the

pulse interval of pulses from the pulse generator, the detectoin error is $1/M_1$. Thus, the detection error can be reduced as compared with the conventional pulse interval counting method. Even if the speed is changed during the set time interval Td, the speed detecting time Td + ΔT is synchronized with the output pulse from the pulse generator, and thus the speed can be detected with high precision.

The fundamental idea of this invention has been described as above. The same thing is true for the method B in Fig. 1. The speed detecting time in method B is Td - ΔT_2.

Although in the methods A and B the set time interval Td is measured in synchronism with the leading edge of the output pulse from the pulse generator, it can be measured in an asynchronous manner as shown in Fig. 1 by C to F. The speed detecting time according to methods C to F are as follows:

Method C	$Td - \Delta T_3 + \Delta T_4$
Method D	$Td - \Delta T_3 - \Delta T_5$
Method E	$Td + \Delta T_6 - \Delta T_5$
Method F	$Td + \Delta T_4 + \Delta T_6$

The basic idea of this invention has been mentioned as above. A specific embodiment of this invention will be described below.

Fig. 2 shows an embodiment of this invention which is applied to a digital control apparatus for motor. Referring to Fig. 2, there is shown a DC motor

4 which is driven by a drive circuit 3. This drive circuit 3 is formed of a power converter having power semiconductors such as thyristors, transistors and so on, and a control circuit for the power converter. A microprocessor 2 is supplied with a speed command value from a speed command circuit 1 and a detected speed value from a speed detecting circuit 6 so as to generate a control signal for controlling the drive circuit 3 to operate. The drive circuit 3 drives the DC motor 4 in accordance with the control signal from the microprocessor 2. A pulse generator 5 generates a pulse signal of a frequency proportional to the frequency of the motor 4.

The operation of the arrangement as shown in Fig. 2 is well known and will not be described. The DC motor 4 is controlled to achieve a speed corresponding to the speed command value.

Fig. 3 shows a specific example of the speed detecting circuit 6 for the method A in Fig. 4, and the speed detecting circuit is represented by 6A. In Fig. 3, a timer 11 is actuated after being set to a time Td by the microprocessor 2. After the time Td, the timer 11 generates a time interruption pulse (hereinafter, referred to as TINT pulse) PT and supplies it to the microprocessor 2. A counter 12 counts the pulse PL in synchronizm with the leading edges of the pulse PL from the pulse generator 5. The contents, MA of the

counter 12 are supplied to the microprocessor 2. A counter 13 counts a clock pulse PC from a clock pulse generator 14, and the contents MB of the counter 13 are supplied to the microprocessor 2. The counters 12 and 13 are reset to zero by a reset pulse PR from the microprocessor 2. A monostable circuit 15 supplies an interruption pulse (hereinafter abbrivated INT) PI to the microprocessor 2 in synchronism with the leading edges of the output pulse PL from the pulse generator 5. The interruption pulse PI is supplied to the microprocessor 2 only when an interruption inhibit pulse (hereinafter, abbriviated NIN pulse) PN is at "1" level, but inhibited from being generated when the NIN pulse PN is at "0" level.

The operation of Fig. 3 will be described with reference to the flowcharts of Figs. 4 to 6, and the timing charts of Figs. 7 to 10.

The microprocessor 2 executes three programs of Figs. 4 to 6 as the processes for speed detection. First, the microprocessor 2 executes the normal process (MAIN process). At step 20 the NIN pulse PN is made to "0" level for inhibition of interruption, and under this speed, speed detection is waited for its start. The speed detection start command, although not shown in Fig. 4, is supplied from a speed control arithmetic process. When the speed detection start command is applied, the program goes to steps 24 and 26 in turn, at which the flags just after the start and for low

speed are set. After the flags are set, the program goes to step 28, where the timer 11 is set to time Td and actuated. Then, at step 30 the reset pulse PR is generated, and the counters 12 and 13 are reset thereby. Subsequently, at step 32 the NIN pulse PN is made "1" level, releasing the interruption from inhibit. This state at step 32 is kept until the detection end command at step 34 is supplied from the speed control arithmetic process. Under this state, the speed detection process is performed as will be described later. When the detection end command is applied, the program goes to step 36, where the NIN pulse PN is made "0" level, and the INT pulse Pl is inhibited from occuring for interruption.

Thus, in the MAIN process of Fig. 4, at step 28, the timer 11 is actuated, and at step 34 the state is kept. Under this condition, when the set time Td comes, the timer 11 generates the TINT pulse PT. In this case, the time interruption process (TINT process) of Fig. 5 is executed. In the time interruption process, first at step 40 the interruption is released from inhibition, and at step 42 the timer 11 is set to time Td and actuated. Then, at step 44 the low speed flag is set as shown in Fig. 7, and at step 46 the contents MB (1) is supplied from the counter 13 to the microprocessor 2. In Fig. 7, the number within the parentheses following the count MB indicates the number of times the detection is made, and MB (n) represents a detected

value at n-th detection. After the count MB (1) is supplied from the counter 13 to the microprocessor 2, the flag just after the start is decided at step 48, and since as shown in Fig. 7 the flag is set, the program goes to step 50. At step 50, it is decided whether the motor 4 is driven or not. The state of this drive is specified by the speed control arithmetic process although not described. If the motor is not being driven, the program goes to step 52, where the speed detected value Nf is made 0. At step 54, the value is stored in a predetermined memory, ending the first process. If the motor is being driven as shown in Fig. 7, the program goes to step 56 where it is decided whether the contents MB (1) of the counter 13 exceeds a constant value M_ℓ or not. If it does not exceed, processes at steps 52 and 54 are executed.

However, if the pulse generator 5 generates no output pulse PL even while the motor is being driven as shown in Fig. 7, the process of Fig. 5 is executed at each time Td. As a result, the counter 13 is not reset, and thus the count MB becomes, for example, MB (3), exceeding the constant value M_ℓ. As a consequence, the program goes to step 58, where the pulse generator 5 is decided to be abnormal.

Thus, if the pulse generator 5 generates no pulse for a predetermined time (the time taken for the count MB of the clock pulse PC to exceed the constant value M_ℓ) from the initiation of speed detection even

while the motor 4 is being driven, the pulse generator 5 is decided to be abnormal. Consequently, a dangerous speed control such as reckless driving is protected from.

When the pulse generator 5 is normal, the speed detection is made as follows. The operation will be described with reference to Fig. 8.

When the motor 4 is driven, the pulse generator 5 generate the output pulse PL. At this time, the microprocessor 2 is in the state at step 32 of Fig. 4, where the interruption is released from inhibit the NIN pulse made "1" level. Thus, the monostable circuit 15 generates the INT pulse PI in synchronism with the leading edges of the pulse PL. The microprocessor 2 executes the interruption process (INT process) of Fig. 6 when supplied with the INT pulse PI.

First, at step 70 the timer 11 is set to constant time Td and at step 72 the interruption within time Td is inhibited. At step 74, the low speed flag is reset, and at step 76 the contents MA and MB of the counters 12 and 13 are supplied to the microprocessor 2. Since the flag just after the start is set in the process using the first INT pulse PI from the start of detection, the program goes to step 80. If the flag just after the start is set, the contents MA and MB of the counters are stored at step 82. When the first INT pulse PI from the start of detection is supplied, the speed detected value Nf is made 0 at

step 84, and stored in a memory at step 86. Thus, in the interruption processing by the first INT pulse PI from the start of detection, the contents MA and MB of the counters 12 and 13 are stored.

When the time Td has elapsed after the application of the first INT pulse PI, the timer 11 generates the TINT pulse PT. The microprocessor 2 executes the time interruption process as shown in Fig. 5 when supplied with the TINT pulse PT.

First, at step 40, the NIN pulse PN is made "1" level releasing the interruption from inhibition, and at step 42 the timer 11 is set to the time Td and actuated. Then, at step 44 the low speed flag is reset in the interruption process by the INT pulse PI, thus the program being progressed to step 60. At step 60, the low speed flag is again set, ending the process.

When the microprocessor 2 is supplied with the INT pulse PI, the program goes from step 78 to steps 88 and 90 since the flag just after the start is already reset. At step 90, the detected value Nf is calculated. This calculation process will be described with reference to the timing chart of Fig. 9. Fig. 9 shows the speed calculation process from the time i at which the INT pulse PI is generated.

When assuming that the contents of the counters 12 and 13 at the generation of the INT pulse PI, or time i are MA (i) and MB (i), respectively, the counts M_1 and M_2 of the pulse PL are calculated at step 88 from the

counts MA (i+1) and MB (i+1) at time (i+1) as

$$M_1 = MA\ (i+1) - MA\ (i) \quad \text{---------- (2)}$$

$$M_2 = MB\ (i+1) - MB\ (i) \quad \text{---------- (3)}$$

At step 90, the speed detected value N_f is determined by substitution of the calculated values M_1 and M_2 from Eqs. (2) and (3) into Eq. (1). The value N_f calculated at step 90 is stored in a predetermined memory at step 86.

At the generation of the INT pulse PI at time i+2, the counts M_1 and M_2 are determined by the equations

$$M_1 = MA\ (i+2) - MA\ (i-1) \quad \text{---------- (4)}$$

$$M_2 = MB\ (i+2) - MB\ (i-1) \quad \text{---------- (5)}$$

and the speed detected value is calculated as described above.

Thus, when the motor 4 is being driven, the process steps 70 to 78, 88, 90, 86 in the interruption process of Fig. 6 and the steps 40, 42, 44, 60 in the time interruption process are repeatedly executed, the speed detected value N_f at each generation of the INT pulse PI can be determined. The speed detected value N_f in the memory at step 86 is updated at each detection of speed, and used for the speed control to the motor 4.

When the motor 4 is being driven at a very low speed, the INT pulse PI occurs at long intervals of time. In this case, the following processes are executed, which will be described with reference to the timing chart of Fig. 10.

It is assumed that the motor 4 rotates at a low speed and the pulse generator 5 generates the output pulse PL as shown in Fig. 10. At the leading edge of the pulse PL, the monostable circuit 15 supplies the INT pulse PI to the microprocessor 2. The microprocessor 2 executes the interruption processing of Fig. 6 and at step 74 the slow speed flag is reset. Under this state, the timer 11 is set, and after time Td has elapsed, the timer 11 supplies the TINT pulse PT to the microprocessor 2. The microprocessor 2 executes the time interruption process as shown in Fig. 5 when supplied with the TINT pulse PT.

First, to release the interruption from the inhibition at step 40, the NIN pulse PN is made "1" level, and at step 42, the timer 11 is set to the time Td and actuated. Then, at step 44, the state of the low speed flag is decided. In this case, since the low speed flag for the interruption process (step 74 in Fig. 6) by the INT pulse PI is reset, the program goes to step 60, where the low speed flag is set. Under this condition, if the INT pulse PI occurs, the speed detection is made as shown in Fig. 9, but if the INT pulse PI does not occur even after the time

Td has elapsed from the generation of the TINT pulse PT, the timer 11 again generates the TINT pulse PT and the TINT process is executed. In this case, since the low speed flag is set by the previous TINT process, the processes at steps 44, 46, 48, 62, 64 are executed. At step 46, the k-th count MB (k) of the counter 13 is supplied to the microprocessor 2. If the count of the counter 13 is MB (j) when the j-th INT pulse PI occurs, the difference, M_{20} = MB (k) - MB (j) is determined at step 62. At step 64, the speed detected value, N_f = K/M_{20} is calculated, and at step 54 the value is stored in a memory, the TINT process being ended.

The above operations are continuously performed until the pulse generator 5 generates the pulse PL. Therefore, calculated values K/Td, K/2Td, K/3Td ..... are obtained in turn at each time Td. Then, when the INT pulse PI occurs, the speed detected value N_f is calculated by the interruption process of Fig. 6. In this case, the speed detection is made as in the conventional pulse interval counting method.

The embodiment of method A in Fig. 1 has been described. It will be understood that the speed can be detected with good resolution and precision even if the speed is suddenly changed. Moreover, even if the speed is reduced to a very low value, substantially the actual speed can be detected at each time interval. Furthermore, since division is made by the software of the microprocessor, the circuit arrangement can be

simplified.

The speed detection by method B in Fig. 1 will be described with reference to Fig. 11, in which like parts as those of Fig. 3 are identified by the same reference numerals. In a speed detecting circuit 6B in Fig. 11, the counter 13 is supplied as a reset pulse, with a logical sum of a reset pulse PR_1 from the microprocessor 2, and a pulse PR_2 which a monostable circuit 102 generates in synchronism with the leading edges of the output pulse PL from the pulse generator 5, through an OR circuit 104. A timer 100 is actuated by the output pulse (INT pulse) PI from the monostable circuit 15, and after lapse of a constant time Td, it generates the TINT pulse PT.

The operations of the arrangement of Fig. 11 will be described with reference to the flowcharts of Figs. 12 and 13 and the timing chart of Fig. 14. The flowcharts of Figs. 12 and 13 show the process for only detecting the speed stationarily, and the start and low speed process described in the embodiment of Fig. 3 are omitted therein.

The microprocessor 2 executes the two processes of the interruption process by the INT pulse PI from the monostable circuit 15 and the time interruption process by the TINT pulse PI from the timer 100. When the monostable circuit 15 generates the INT pulse PI, the microprocessor 2 executes the interruption process of Fig. 13. First, at step 110, the contents MA (i) of the

counter 12 is supplied to the microprocessor 2. Since the leading edges of the pulse PL at which the counter 12 counts up occur before the monostable circuit 15 generates the INT pulse PI, the count MA (i) supplied in the INT process includes the pulse PL at the time of generation of the INT pulse PI. At step 112, the NIN pulse PN is made "0" level, inhibiting the INT pulse PI from generation.

On the other hand, the INT pulse PI is supplied to the timer 100 as a triger signal thereto. The timer 100 generates the TINT pulse PI the time Td after the INT pulse PI is supplied to the timer 100. The microprocessor 2 executes the TINT process of Fig. 15 when supplied with the TINT pulse.

First, at step 114, the microprocessor 2 receives the counts MA (i+1) and MB (i+1) from the counters 12 and 13 at time (i+1) at which the TINT pulse occurs. At step 116, the count MA (i) received at time i in the INT process, the above counts MA (i+1) and MB (i+1) are used for the calculation of the variation M_1 of the pulse PL from

$$M_1 = MA\ (i+1) - MA\ (i) \quad \text{----------} \quad (6)$$

At step 118, the NIN pulse PN is made "1" level to release the interruption in the INT process from inhibition, and at step 120, the value, M_2 is calculated from

$$M_2 = MB_T - MB\ (i+1) \quad \text{----------} \quad (7)$$

where MB_T represents the count of the output clock pulse PC from the clock pulse oscillator 14 for time Td, and is a constant value proportional to the time Td. The MB (i+1) is a value proportional to the time ΔT_2 between the end point (i+1) of the time Td and the leading edge of the pnevious PL just therebefore. Thus, the value M_2 determined by Eq. (7) is a value proportional to the difference, Td - ΔT_2.

At step 122, Eq. (1) is calculated by substituting the values M_1 and M_2 obtained from Eqs. (6) and (7), and at step 124 the speed detected value N_f is stored in a memory to end the TINT process. The INT process and the TINT process as described above are repeatedly performed to detect the speed.

Also in the embodiment of Fig. 11, the speed detected value is obtained with high resolution. Moreover, since the speed detecting time is constant, or Td, the algorithm of the speed control computation having a relatoin with time, for example, the process using integrating compensation can be performed simply.

Fig. 15 shows another example of the speed detecting circuit of this invention. In this arrangement, the start point of the set time Td is not in synchronism with the output pulse PL from the pulse generator 5 for detection of the speed. In order to detect the speed with the start point of the set time

Td being not in synchronism with the pulse PL, there are employed methods C to F in Fig. 1. The method D will be described below.

The arrangement of Fig. 15 is different from that of Fig. 3 in that a timer 200 in a speed detecting circuit 6C generates the TINT pulse PT at each time Td. The time interruption process by the TINT pulse PT from the timer 200 has a priority lower than that of the interruption process by INT pulse PI. Therefore, the microprocessor 2 interrupts the execution of the time interruption process and executes the interruption prior thereto when supplied with the INT pulse PI.

The operations of the arrangement of Fig. 15 will be described with reference to the flowcharts of Figs. 16 and 17 and the timing chart of Fig. 18.

In Fig. 15, the microprocessor 2 performs the calculation of the speed detected values in the time interruption process and it is supplied with data necessary for the time interruption process and executes the preliminary computation in the interruption process.

When the timer 200 generates the TINT pulse PT at n-time point, the microprocessor 2 executes the TINT process as shown in Fig. 16. First, at step 202, the NIN pulse PN is made "0" level, inhibiting the INT pulse PI from interrupting, and at step 204, the counts MA (n) and MB (n) of the counters 12 and 13 are supplied to the microprocessor 2. At step 206, the flag is set which decides that the first INT pulse PI

has been generated after the INT pulse PT occured. At step 208, the NIN pulse is made "1" level, releasing the interruption from the inhibition. At steps after step 208, the TINT process is executed, but when the INT pulse PI occurs, the INI process of Fig. 17 is executed. For convenience of explanation, it is assumed that the process 210 and the followings are performed continuously in turn. At step 210, the count, MB n-1(k) at time n-1(k) at which the INT pulse PI occurs just before n-time point is subtracted from the count MB (n) of the counter 13. The generation time point n-1(k) is a time point at which the k-th pulse PL occurs after the TINT pulse PT was generated at n-1 time point. The value, ΔMB_2 (n) determined at step 210 is the time interval between the TINT pulse PT occuring at n-time point and the leading edge of the pulse PL generated just therebefore, and is proportional to time ΔT_5 in Fig. 1. At step 212, the value M_1 is determined by subtracting 1 of pulse PL from the difference between the count MA (n) of the counter 12 at n-time point and the count MA (n-1) at time-point n-1. The subtraction of 1 pulse PL is necessary because the count MA (n-1) at (n-1)-time point includes a value of 1 which is counted out of the set time Td. At step 214, the value M_2 is determined by substituting the counts MB (n) and MB (n-1) of the counter 13, the value ΔMB_2 (n) at step 210, and ΔMB_1 (n-1) determined by the INT process which will be described later, into

the equation (8),

$$M_2 = MB(n) - MB(n-1) - \Delta MB_2(n) - \Delta MB_1(n-1) \quad \text{---------- (8)}$$

The value M_2 obtained from Eq. (8) is a value proportional to the time Tdo as shown in Fig. 18. At steps 216, the speed N_f is calculated by using the values M_1 and M_2, and at step 217 the speed detected value N_f is stored.

On the other hand, when supplied with the INT pulse PI, the microprocessor 2 executes the INT process of Fig. 17. First, at step 218, the count MB of the counter 13 is supplied to the microprocessor 2, and at step 220, decision is made of the state of the flag of the INT pulse PI. The flag of the INT pulse is set by the TINT process of Fig. 16 if the INT pulse PI is the first one after the TINT pulse PT was generated. If the INT pulse is the first one after the TINT pulse PT was generated at n-time point, the flag is set, and at step 222 the value ΔMB_1 (n) is determined from the equation (9),

$$\Delta MB_1 (n) = MB\ n(1) - MB\ (n) \quad \text{------- ------ (9)}$$

The ΔMB_1 (n) in Eq, (9) is a value proportional to the time ΔT_3 in method D in Fig. 1. This value ΔMB_1 (n) is stored for use in the calculation of speed by the

TINT pulse at time n+1. For the calculation of speed at n-time point is used the difference value MB_1 (n-1) between the counts of the counter 13 at the TINT pulse of time n-1 and the first INT pulse PI just thereafter.

When the process at step 222 is completed, the flag for INT pulse is reset at step 224, and then the count MB = MBn(k) of the counter 13 is stored at step 226. In this case, MB (n) becomes MB (1). Thereafter, the INT processes up to occurence of the TINT pulse PT is performed at steps 220 to 226 in turn, and at the generation of the INT pulse, the count of the counter 13 is stored.

Thus, even if the start point of the set time Td is not synchronized with the output pulse PL from the pulse generator 5, the speed can be detected. Thus, there is no need to synchronize the start point of the set period Td with the pulse PL and the speed detection can be performed with a simple arrangement.

While the method D in Fig. 1 has been described with reference to Fig. 15, it will be evident that the method C can be implemented easily. The method C will not be described for the sake of convenience.

Moreover, the methods E and F in Fig. 1 can be executed likewise by determining the values M_1 and M_2 in the embodiment of Fig. 15 from the following expression;

for method E

$$M_1 = MA\ (n) - MA\ (n-1) \quad \text{----------} \quad (10)$$

$$M_2 = MB\ (n) - MB\ (n-1) - \Delta MB_2\ (n) + \Delta MB_2\ (n-1) \quad \text{---------- (11)}$$

1 for method F

$$M_1 = MA\ (n) - MA\ (n-1) \quad \text{---------- (12)}$$

$$M_2 = MB\ (n) - MB\ (n-1) + \Delta MB_1\ (n) + \Delta MB_2\ (n-1) \quad \text{---------- (13)}$$

Further explanation thereof will be omitted.

In the above embodiment, the speed detecting time is substantially constant as time Td, and in the steady state in which the motor speed does not almost change, it is necessary to prolong the speed detecting time to improve the precision of speed detection.

The extension of the speed detecting time can be performed for example by the INT process in the embodiment of Fig. 3 and the addition of the process of Fig. 19. Specifically, the process of Fig. 19 is added between the steps 90 and 86 in Fig. 6.

The operations of this case will be described with reference to the timing chart of Fig. 20. At step 90, the speed detected value $N_f\ (m)$ is computed, whrere m represents a number of order. At step 132, the previous detected value $N_{fo}\ (m-1)$ is subtracted from the m-th value. The final speed detected value is represented by N_{fo}, and N_f represents the result obtained by the computation at step 90. If $|N_f(m) - N_{fo}(m-1)|$

exceeds a preselected speed change setting value ΔN_o, the step 134 and the followings are performed. The value ΔN_o is selected to be desirably about the maximum value of the variation of the speed detected value N_f measured at each $Td + \Delta T_1$ when the motor is rotated at a constant speed.

When the difference between the previous value and this value exceeds ΔN_o, steps 134 to 138 are executed and the speed detected value N_f obtained at step 90 is stored as N_{fo}. The $\ell=1$, and SM_1, SM_2 at steps 134 and 136 are set for the process which will be described later. When the motor speed N is changed as shown in Fig. 17, detected values up to the value N_f (m-1) are processed at steps 134 to 138.

If the $|N_f (m) - N_{fo} (m-1)|$ is reduced to less than ΔN_o at the detectoin of N_f (m) in Fig. 17, steps 140 and the followings are performed. If, now, ℓ_o of 5 is established, at step 134, ℓ becomes 1, and thus the program goes to step 142. At step 142, ℓ is made 2 and SM_1 and SM_2 are calculated at steps 144 and 146. Since the SM_1 and SM_2 include M_1 (m-1) and M_2 (m-1) at time m-1 at step 136, the new SM_1 and SM_2 are the sum of the second counts and those values. In other words, the detecting time is extended to about 2Td. At step 148, the speed detected value N_{fo} is calculated. Similarly, for ℓ = 2, 3, 4, the detecting time is extended to 3Td, 4Td, 5Td, respectively thus detection precision being improved. For ℓ = 5, the program goes to step 150, and steps 152 and 154 are executed. This is because the detecting time is limited to ℓ_oTd (here, 5Td). Therefore, the counts

M_1 $(m-\ell_o)$, M_2 $(m-\ell_o)$, ℓ_o times before are subtracted from the counts M_1 (m), M_2 (m) at this time and values M_1 (m), M_2 (m) are added thereto, respectively. The speed detected values N_{fo} (m+4) and so on at time m+4 and the followings are obtained at steps 150 to 154.

As described above, in the steady state in which the speed is not almost changed, the speed detecting time is extended thereby improving the precission of detectoin. In this case, when the speed is suddenly changed, the program goes to steps 134 to 138, and therefore the response for the speed detection is never lost.

Although in the description with reference to Fig. 19, the speed detecting time is extended when the change of the speed detected value is below a preset value, the detection precision can be improved even if the speed detecting time is extended when the speed detected value is small or when the speed control devision is small. In that case, the process at step 132 is designed to be each algorithm.

The improvement of the detection precision by the extension of the speed detecting time can similarly be achieved by the methods B to F as well as method A.

Fig. 21 shows another example of the speed detecting circuit of this invention, in which the speed is detected each time a pulse generator produces an output pulse.

The arrangement of Fig. 21 is different from that of Fig. 3 in that a speed detecting circuit 6 D has no

timer 11.

The operations of the arrangement of Fig. 21 will be described with reference to the flowcharts of Figs. 22 and 23 and the timing chart of Fig. 24.

The microprocessor 2 treats the MAIN process as shown in Fig. 22 and the INT process in Fig. 23. The MAIN process comprises steps 156 to 166 for chiefly deciding whether or not the detection is started, while the INT process is that steps 170 to 196 are executed in synchronism with the output pulse from the pulse generator 5 and after time Δt, the program returns to the MAIN process.

In the MAIN process, first, the MIN pulse is made "0" level in order to inhibit the interruption at step 156. Then, at step 158, decision is made of the state of the detection start command. The detection start signal is stored at a certain address of a memory in the micro-processor 2. When the detection start command is to start detection at level "1", the program goes to step 160. When the detection must not be started at "0" level, the process at step 158 is continuously performed until the detection start signal becomes "1" level. At step 160, the flag just after the start of detection is set. Then, at step 162, the interruption is released from the inhibition, or the NIN pulse PN is made "1" level,. making the monostable circuit 15 operable. Thereafter, at step 164, decision is continuously made of the end of detection until the detection end signal is generated.

Under the execution of step 164, the output pulse PL from the pulse generator 5 is applied to the monostable circuit 15 and the INT pulse PI is applied to the microprocessor 2, which thus executes the INT process.

In the INT process, first, at step 170 the interruption is inhibited so that when the processing in the microprocessor 2 is slow, the next INT pulse PI when generated is prevented from being of force during the execution of INT process. Then, the state of the flag just after start is decided at step 172. When the INT process is synchronized with the pulse PL generated at time t_o just after start of detection, the program goes to step 174, where the flag just after start of detection is reset for the execution of the following steps 172 to 182. After the step 174 is executed, the program goes to step 176, where the microprocessor 2 receives the counts MA (0) and MB (0) of the counters 12 and 13, respectively and stores them in its memory. Here, MA (0) and MB (0) are counts at time t_o and MA (n) and MB (n)are counts at time t_n (see the counts MA and MB in Fig. 24).

At steps 178 and 180, $k = Q$ and $N_f = 0$ are established. At t_o, no speed detected value is obtained. After step 180 is finished, step 194 is executed to store the detected value N_f in a predetermined memory. Of course, when the value is fed to an external apparatus, the value may be digital to the apparatus. At step 196, the interruption is released from the inhibition in order to treat the next INT process.

In the second INT process and the follewings from time t_1, steps 182 to 192 are executed. Now in the n-th process (INT process at time t_n), at step 182 the counts MA (n)and MB (n) of the counters 12 and 13 are supplied to the microprocessor 2, and stored in a certain memory thereof. Then, at step 184, decision is made of whether the difference between the count MB (n) of the counter 13 at this time and the count MB (k) thereof at time t_k before time t_n exeeds a constant value MB_T or not, where constant value MB_T is the number of clock pulses generated during the time Td. Therefore, at step 184 decision is made of whether the time $(t_n - t_k)$ exceeds time Td or not. When time $(t_n - t_k)$ exceeds time Td, the program goes to step 186, where k is incremented by 1. Then, the program goes to step 184, again. Until the MB (n) - MB (k) - MB_T becomes negative, the loop of steps 184 and 186 is repeatedly executed, and when it becomes negative, the program goes to step 188, where decision is made of k = 0. When k = 0 (which corresponds to time t_1, t_2), there is only the first detected value, and thus the program goes to step 192 for calculating the speed from the counts MA (0) MB (0) at that time. When $k \neq 0$, the obtained $\underline{k}$ indicates the address at which data preceding by time Td or above from the present time t_n and positioned at time t_k nearest to time Td is stored. At step 184, if the MB (n)- MB (k) - MB_T = 0, the time t_k satisfies the relationship of t_n - Td = t_k and thus immediately the step 192 is executed.

At step 192, the following equation of

$$N_f = K_4 \frac{MA\ (n) - MA\ (k)}{MB\ (n) - MB\ (k)} \quad \text{----------} \quad (14)$$

is calculated by substituting thereinto the values MA (k), MB (k) stored at the addresses specified by k in each process and the values MA (n), MB (n) at this time.

The detected value N_f is stored in a predetermined memory at step 194, and the interruption is released from the inhibition for the next INT process at step 196. Such operations are performed each time the pulse PL occurs, and the speed detected value N_f is calculated. For example, in the timing chart of Fig. 24, the average speed in the interval T (1), that in the interval T (2), and that in the interval T (n) can be detected at time points t_1, t_2 and t_n, respectively. Of course, T (n)is near the set time Td.

As described above, in this embodiment, since the speed can be detected each time the pulse generator generates an output pulse, there is an effect of reducing the speed detection delay. Moreover, since the microprocessor 2 calculates the speed in synchronism with the output pulse PL from the pulse generator 5, the hardware arrangement is very simple. Furthermore, since the microprocessor 2 performs the interruption inhibiting process before starting to calculate the speed, the interruption during calculation is inhibited when the period with which the pulse PL is generated becomes

shorter than the processing time Δt in the microprocessor 2. Therefore, there is an effect of causing no error in the speed detected value.

Thus in accordance with this invention, the speed detection using the output pulse from the pulse generator can be performed with good resolution and precision even if the speed changes.

WHAT IS CLAIMED IS:

1. A speed detecting method of detecting the speed of a vehicle by counting the output pulse from a pulse generator which generates a pulse each time said vehicle moves by a predetermined distance, wherein the number of pulses for detection of the speed of said vehicle is counted during the time interval from when said pulse is generated at the measuring start point of a set time interval, or just before or after the measuring start point, to when said pulse is generated just before or after the end point of said set time interval.

2. A speed detecting method accordidng to Claim 1, wherein the start point of said set time interval is synchronized with the generation of pulse from said pulse generator.

3. A speed detecting method according to Claim 1, wherein said set time interval can be changed in length in accordance with the speed of said vehicle.

4. A speed detecting method comprising steps of:

counting an output pulse which a pulse generator produces each time a vehicle moves by a predetermined distance;

measuring a time interval from when said pulse is generated at the measuring start point of a set time interval or just before or after said measuring start point, to when said pulse is generated just before or after the end point of said set time interval; and

calculating the speed of said vehicle from the

measured time interval and the number of the output pulses from said pulse generator which are counted during said measured time interval.

5. A speed detecting apparatus comprising a pulse generator (5) for generating a pulse each time a vehicle moves by a predetermined distance;

counting means (12) for counting the output pulses from said pulse generator;

time measuring means (11) for measuring a time interval from when said pulse is generated at the measuring start point of a set time interval or just before or after the measuring start time point, to when said pulse is generated just before or after the end point of said set time interval; and

speed calculating means (2) for calculating the speed of said vehicle from the count of said counting means which count is a value during the time interval measured by said time measuring means.

6. A speed detecting apparatus comprising:

a pulse generator (5) for generating a pulse each time a vehicle moves by a predetermined distance;

first counter means (12) for counting the output pulses from said pulse generator;

second counter means (13) for counting a clock pulse of a constant repetition frequency higher than the frequency of the output pulse from said pulse generator; and

speed calculating means (2) for determining the

speed of said vehicle from the count of said second counter means which count is a value during a time inteval from when said pulse is generated at the measuring start point of a set time interval or just before or after the measuring start time point to when said pulse is generated just before or after the end point of said set time interval.

7. A speed detecting apparatus according to Claim 6, wherein said speed calculating means is formed of a microprocessor.

FIG. 1

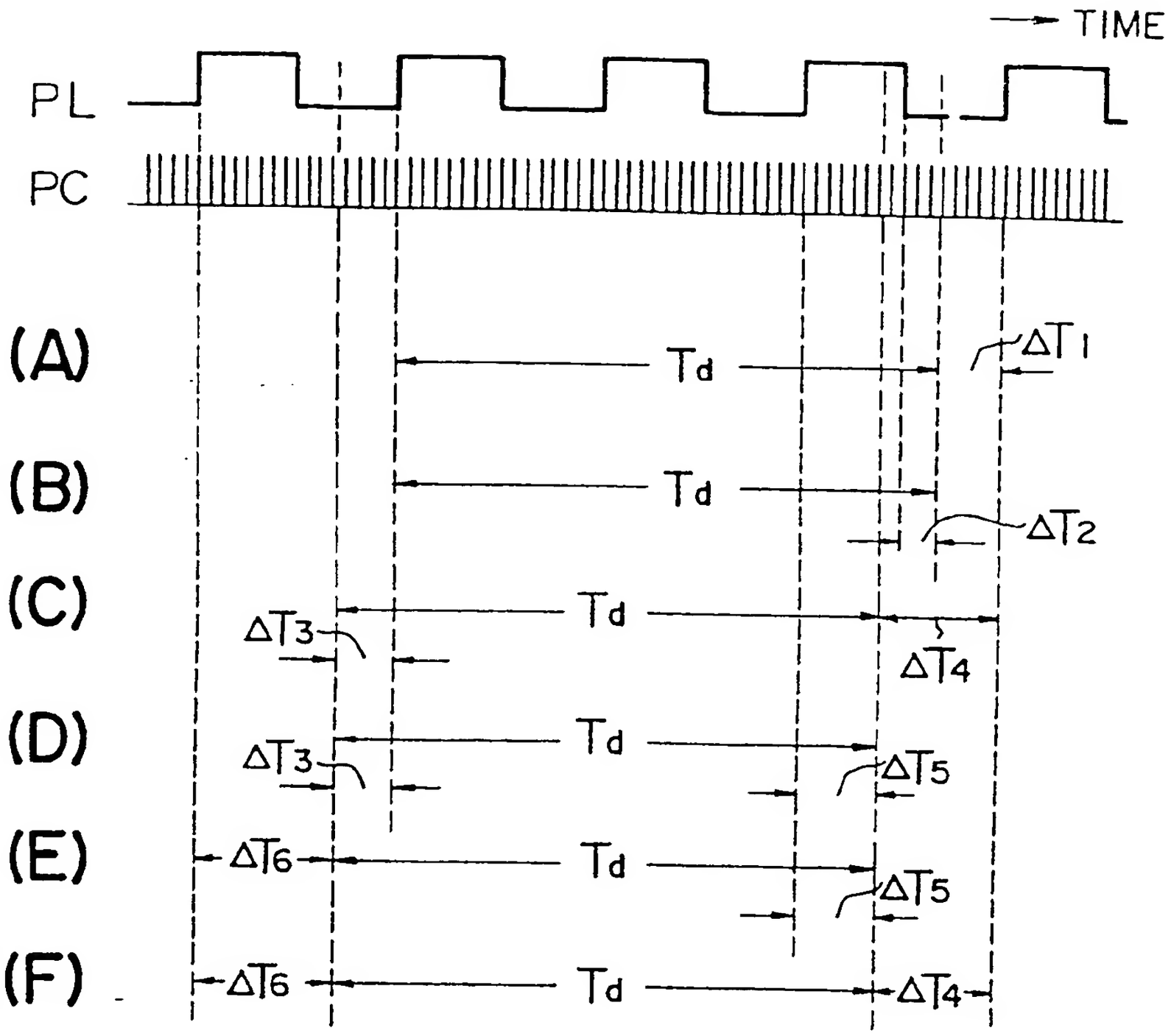

TIME
PL
PC
(A)
(B)
(C)
(D)
(E)
(F)
Td
ΔT1
ΔT2
ΔT3
ΔT4
ΔT5
ΔT6

FIG. 2

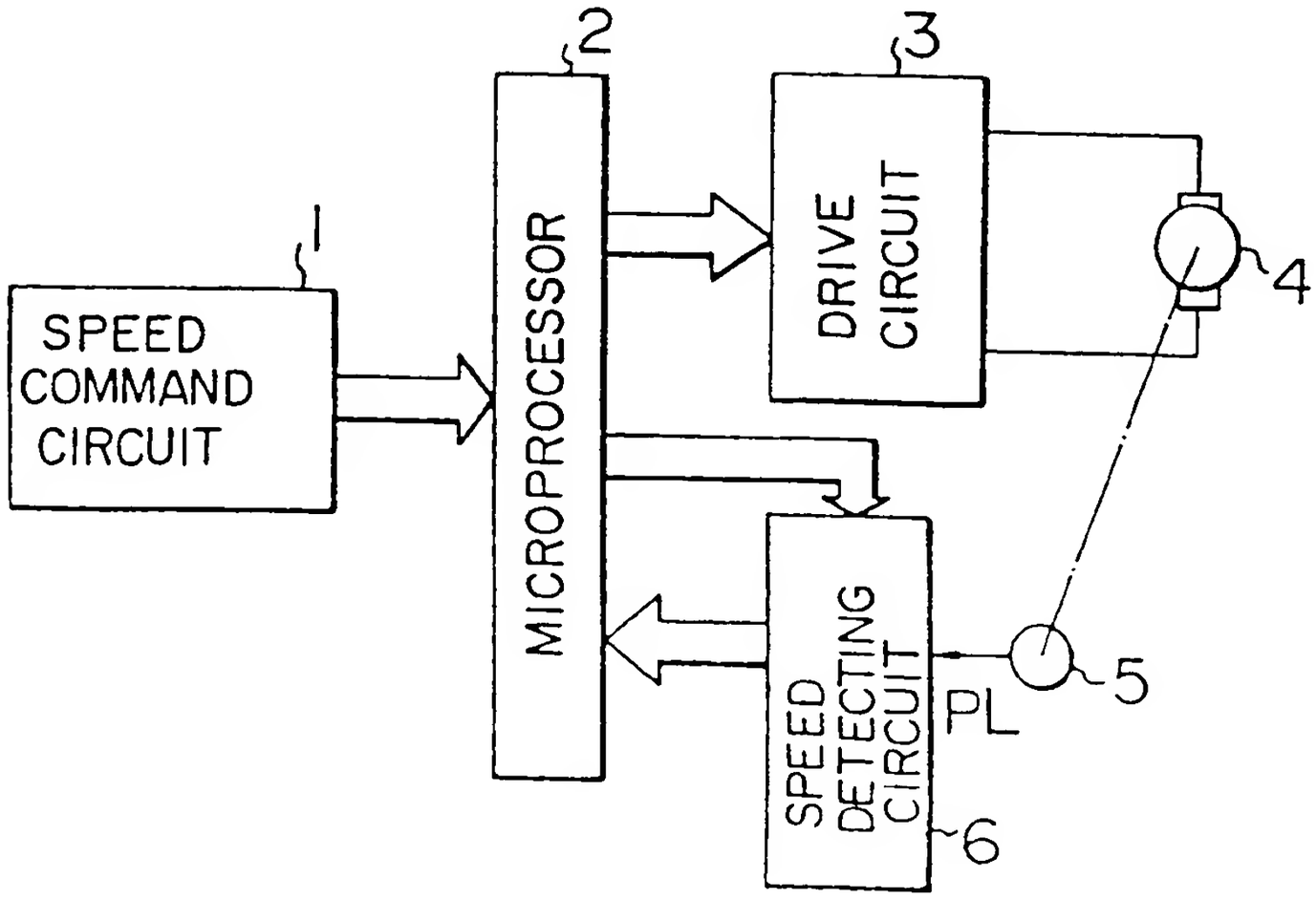

SPEED COMMAND CIRCUIT
1
MICROPROCESSOR
2
DRIVE CIRCUIT
3
4
SPEED DETECTING CIRCUIT
5
6
PL

FIG. 3

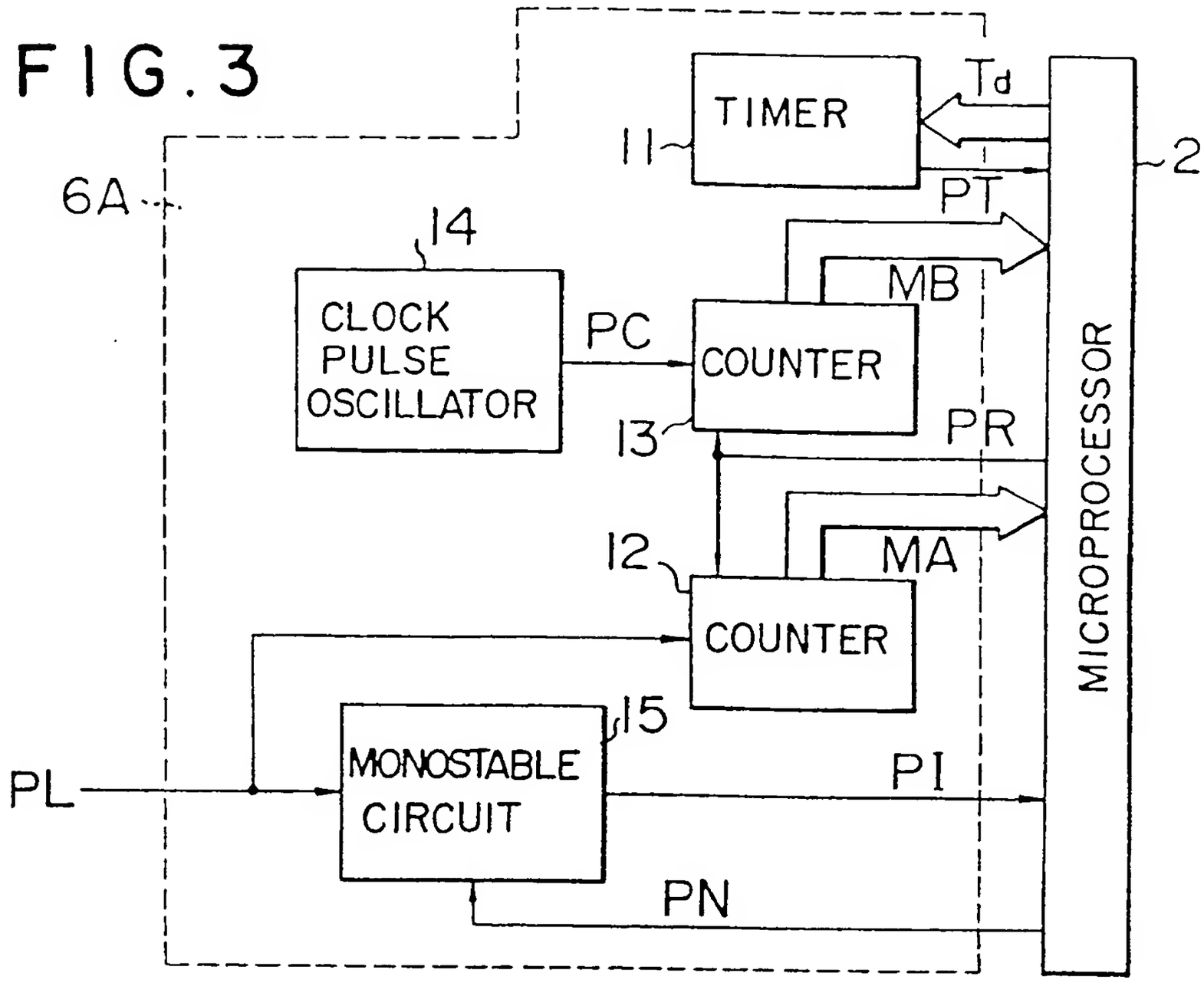

6A
TIMER
11
Td
2
PT
14
MB
CLOCK PULSE OSCILLATOR
PC
COUNTER
13
PR
12
MA
COUNTER
MICROPROCESSOR
15
MONOSTABLE CIRCUIT
PL
PI
PN

FIG. 4

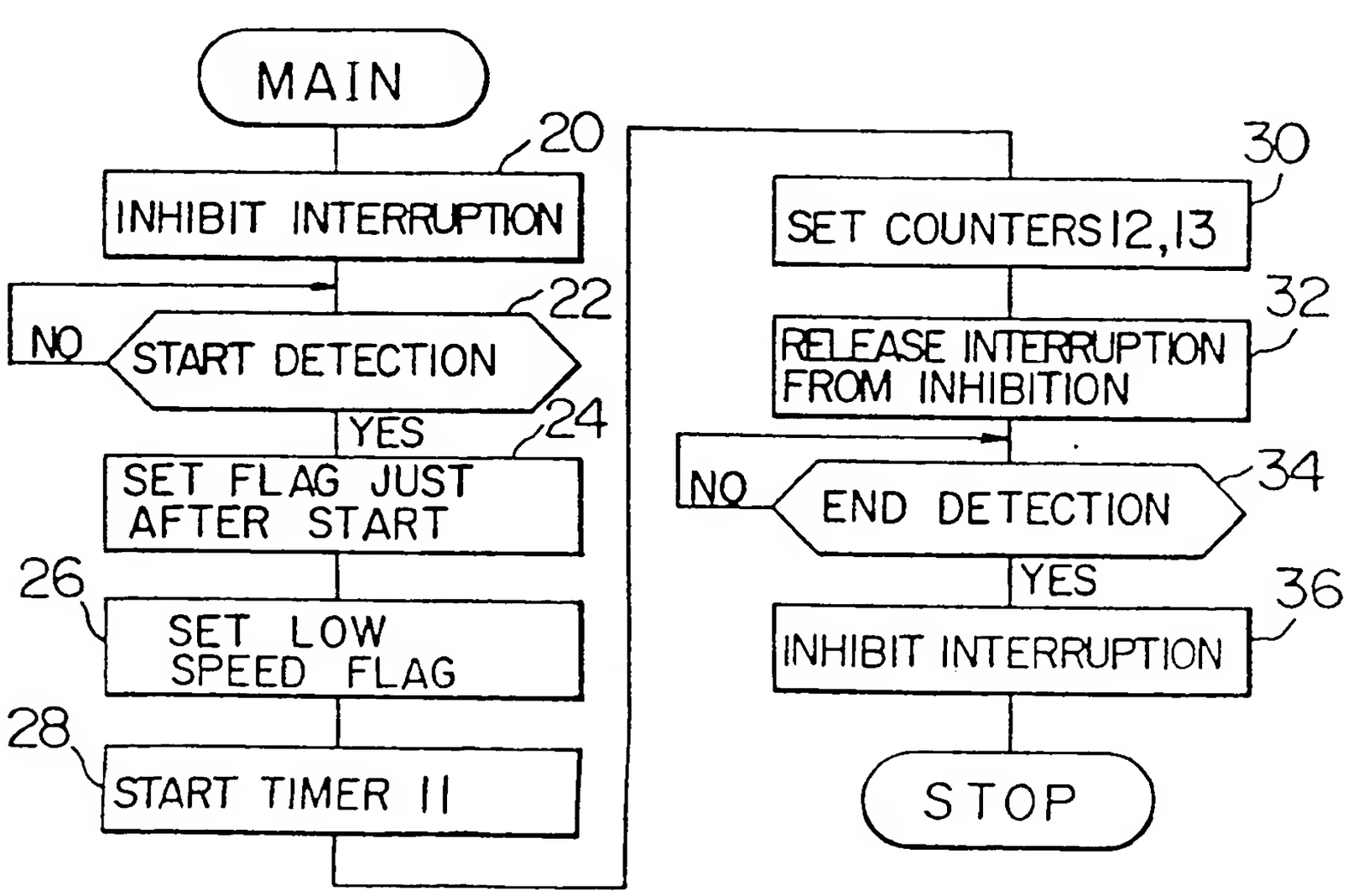

MAIN
20
INHIBIT INTERRUPTION
22
NO
START DETECTION
YES
24
SET FLAG JUST AFTER START
26
SET LOW SPEED FLAG
28
START TIMER 11
30
SET COUNTERS 12,13
32
RELEASE INTERRUPTION FROM INHIBITION
34
NO
END DETECTION
YES
36
INHIBIT INTERRUPTION
STOP

FIG. 5

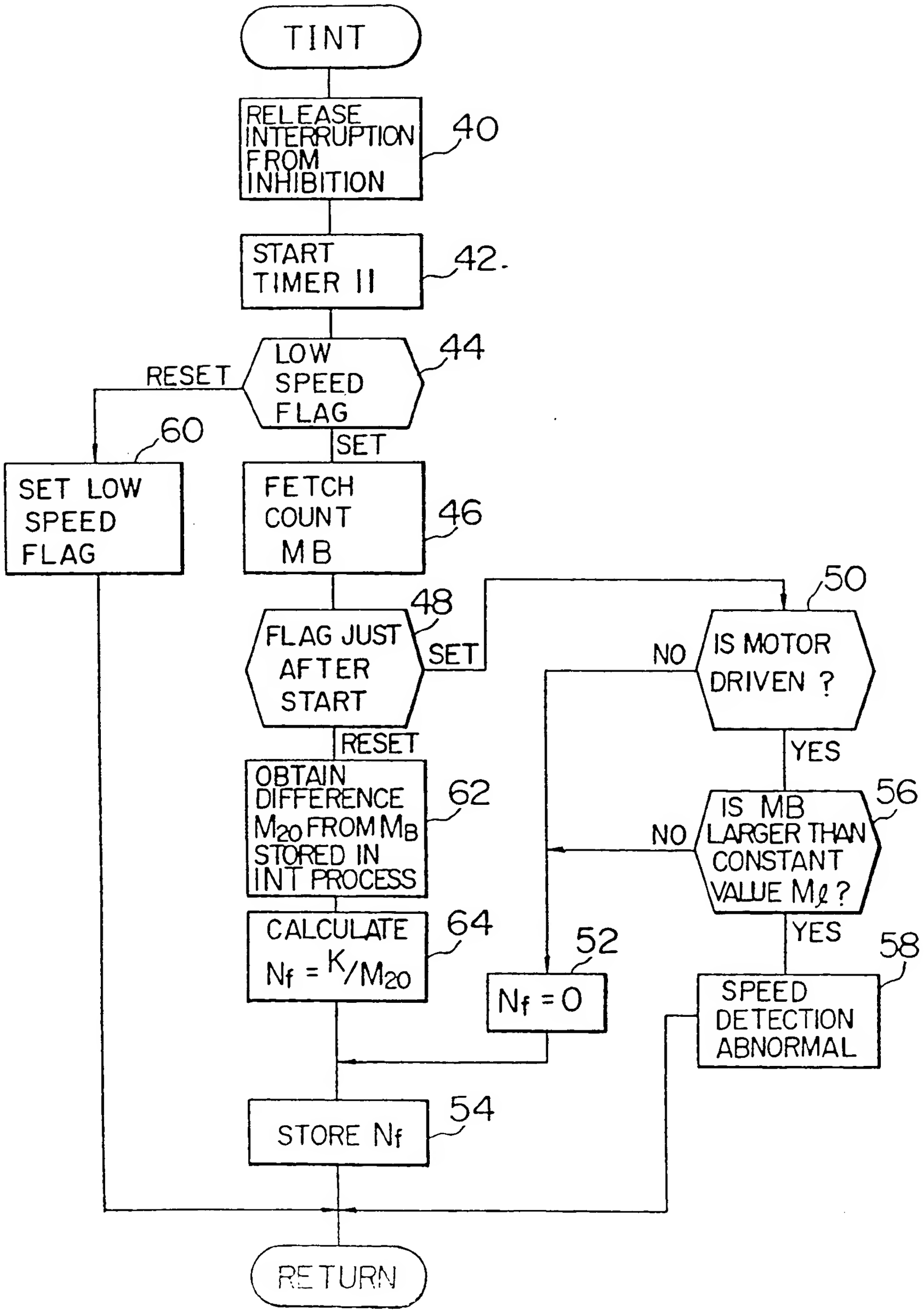

TINT
RELEASE INTERRUPTION FROM INHIBITION
40
START TIMER II
42
LOW SPEED FLAG
44
RESET
SET
60
SET LOW SPEED FLAG
FETCH COUNT MB
46
48
FLAG JUST AFTER START
SET
RESET
50
NO
IS MOTOR DRIVEN ?
YES
OBTAIN DIFFERENCE M_{20} FROM M_B STORED IN INT PROCESS
62
IS MB LARGER THAN CONSTANT VALUE M_ℓ ?
56
NO
YES
CALCULATE $N_f = K/M_{20}$
64
52
$N_f = 0$
58
SPEED DETECTION ABNORMAL
STORE N_f
54
RETURN

FIG. 6

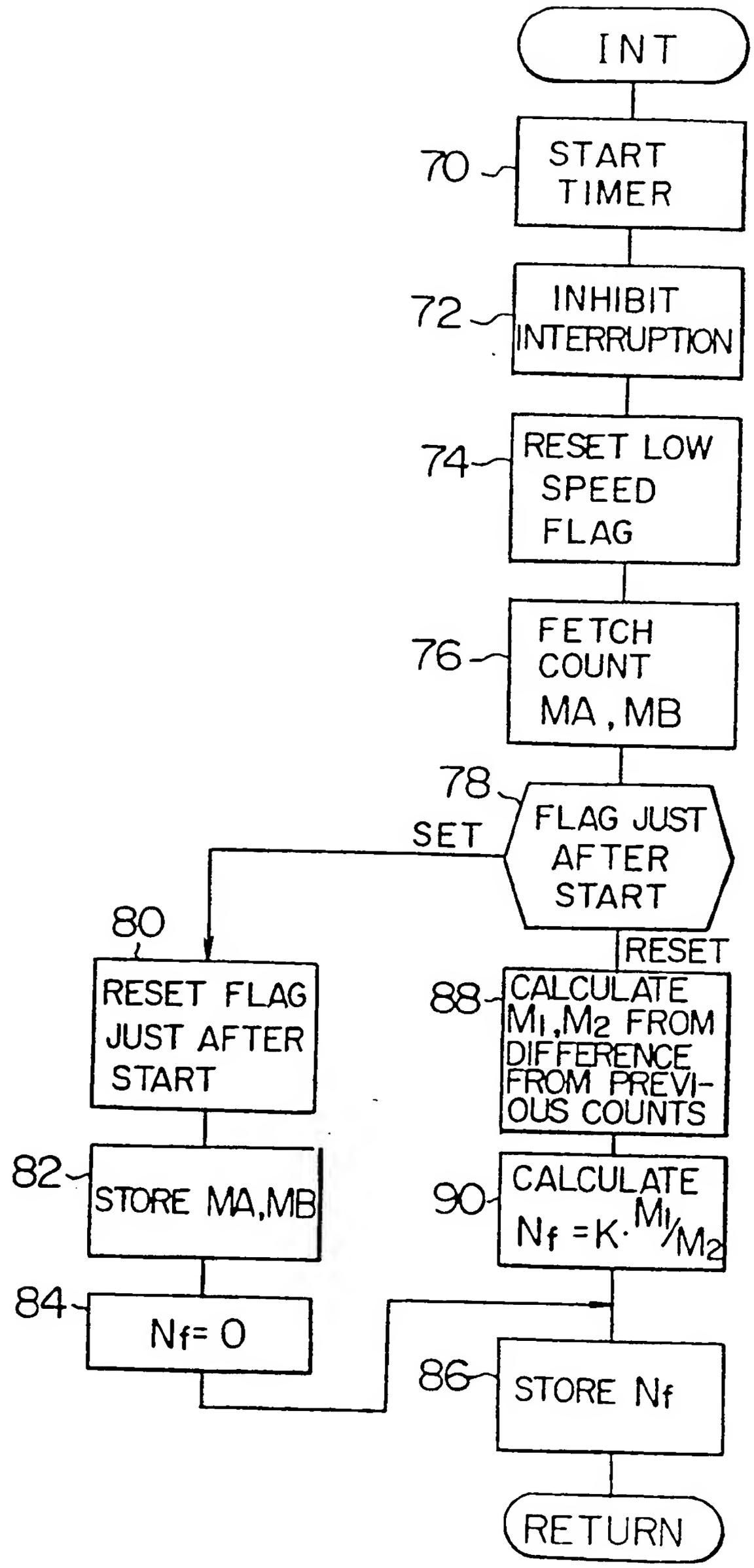

INT
70
START TIMER
72
INHIBIT INTERRUPTION
74
RESET LOW SPEED FLAG
76
FETCH COUNT MA, MB
78
FLAG JUST AFTER START
SET
RESET
80
RESET FLAG JUST AFTER START
82
STORE MA, MB
84
Nf = 0
88
CALCULATE M1, M2 FROM DIFFERENCE FROM PREVIOUS COUNTS
90
CALCULATE Nf = K · M1/M2
86
STORE Nf
RETURN

FIG. 7

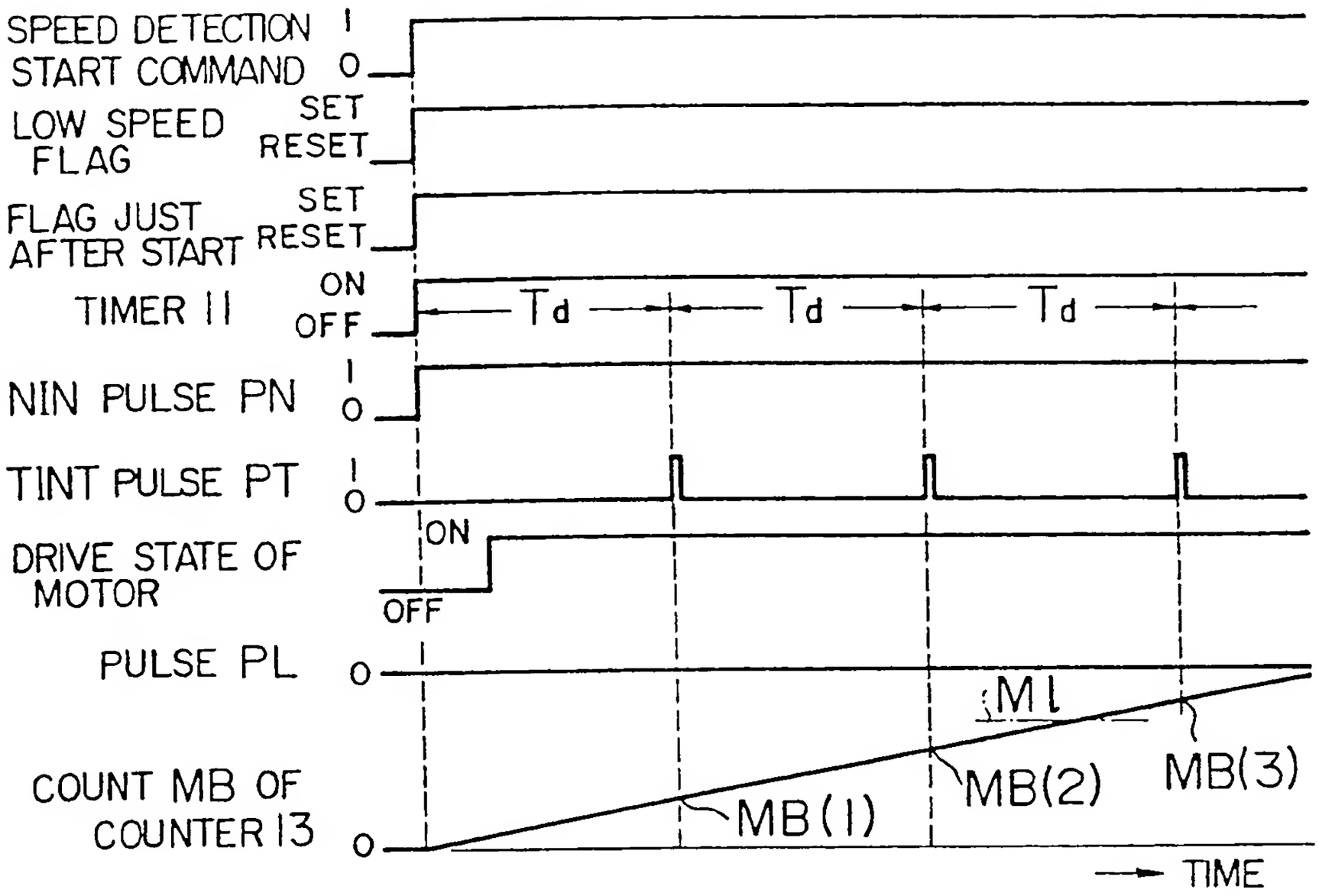

SPEED DETECTION START COMMAND
I
O
LOW SPEED FLAG
SET
RESET
FLAG JUST AFTER START
SET
RESET
TIMER 11
ON
OFF
Td
Td
Td
NIN PULSE PN
I
O
TINT PULSE PT
I
O
DRIVE STATE OF MOTOR
ON
OFF
PULSE PL
O
M1
COUNT MB OF COUNTER 13
O
MB(1)
MB(2)
MB(3)
TIME

FIG. 8

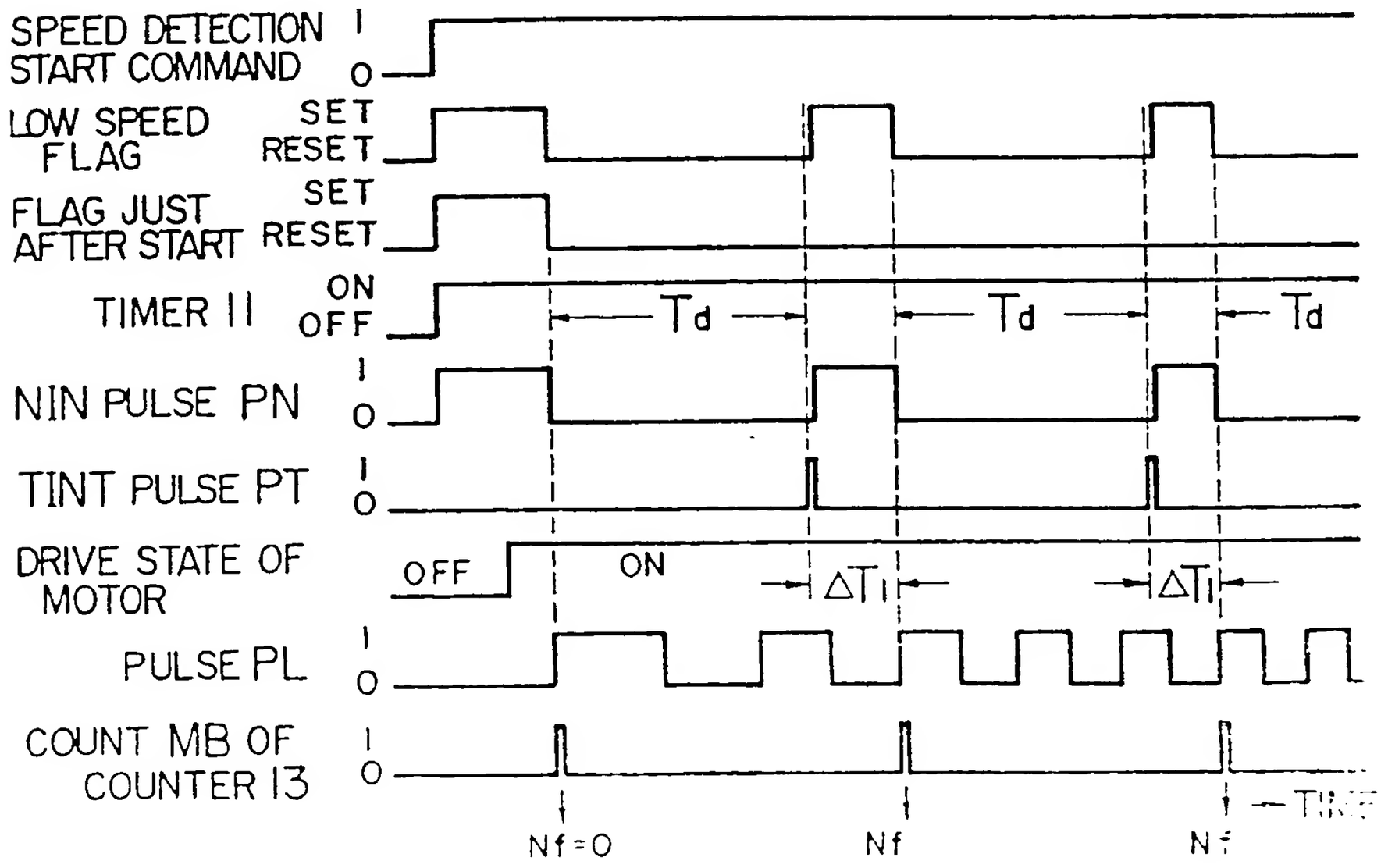

SPEED DETECTION START COMMAND
I
O
LOW SPEED FLAG
SET
RESET
FLAG JUST AFTER START
SET
RESET
TIMER 11
ON
OFF
Td
Td
Td
NIN PULSE PN
I
O
TINT PULSE PT
I
O
DRIVE STATE OF MOTOR
OFF
ON
ΔTi
ΔTi
PULSE PL
I
O
COUNT MB OF COUNTER 13
I
O
Nf=0
Nf
TIME

FIG. 9

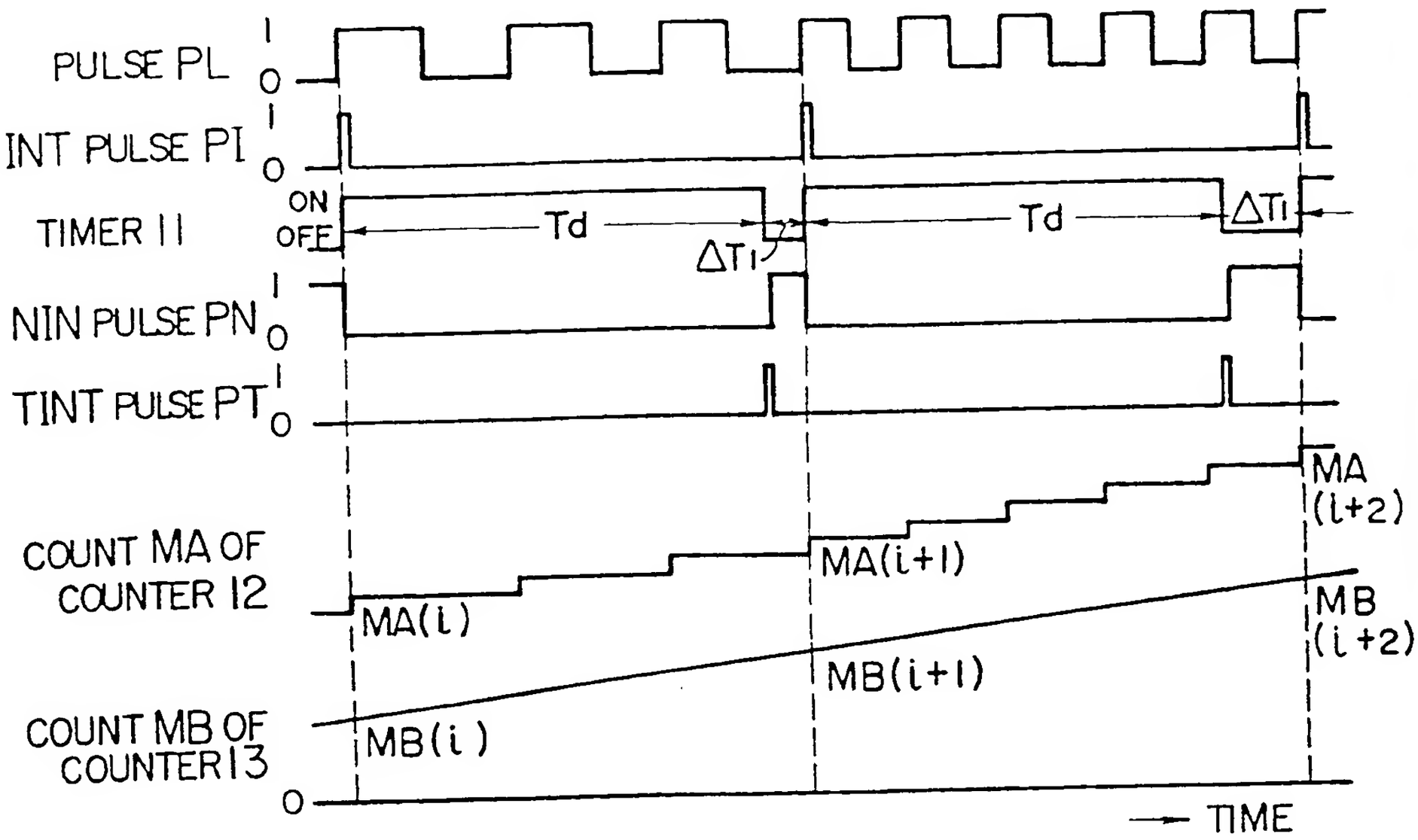

PULSE PL
INT PULSE PI
TIMER 11
ON
OFF
Td
Td
ΔTi
ΔTi
NIN PULSE PN
TINT PULSE PT
COUNT MA OF COUNTER 12
MA(i)
MA(i+1)
MA (i+2)
COUNT MB OF COUNTER 13
MB(i)
MB(i+1)
MB (i+2)
0
TIME

FIG. 10

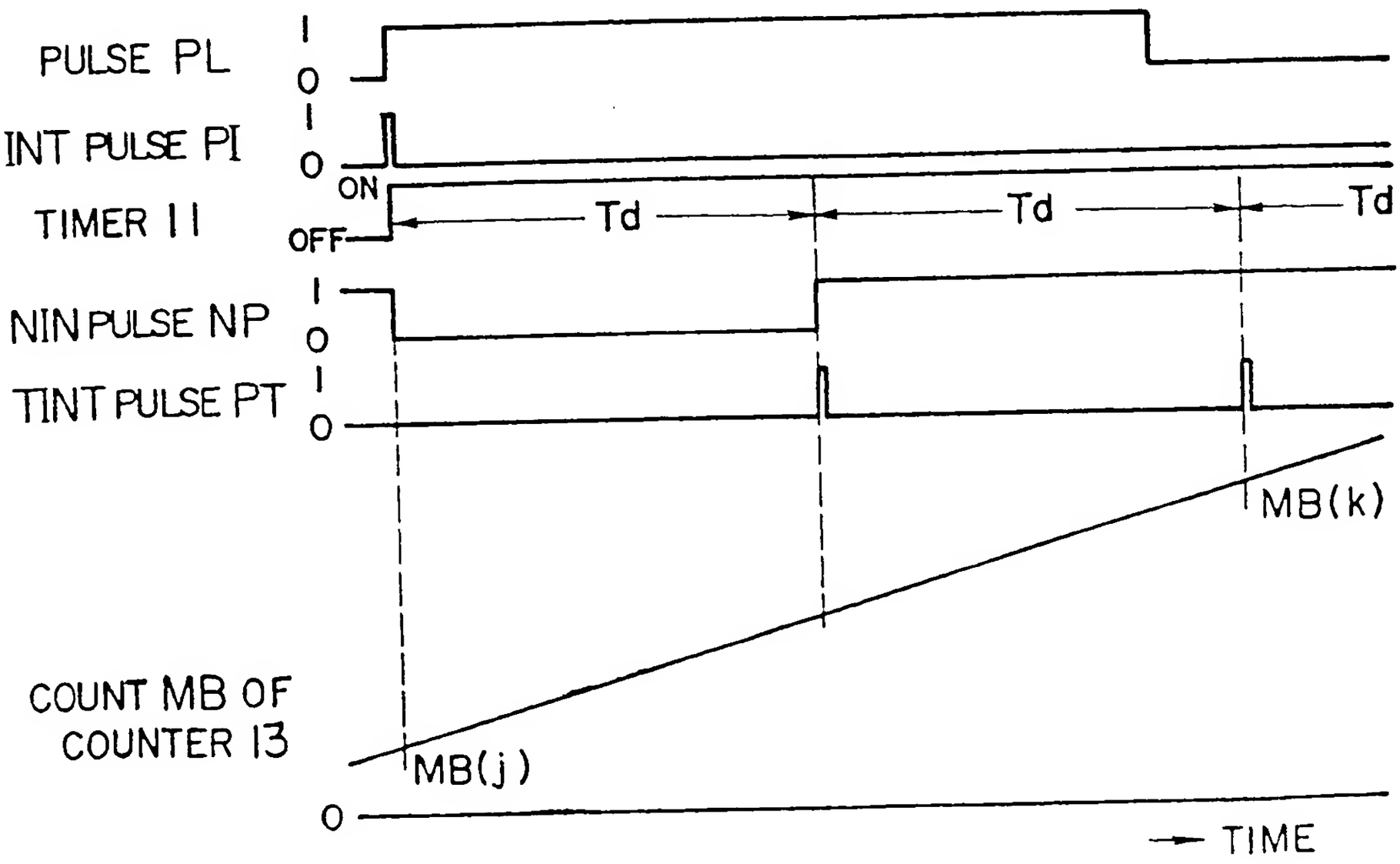

PULSE PL
INT PULSE PI
TIMER 11
ON
OFF
Td
Td
Td
NIN PULSE NP
TINT PULSE PT
COUNT MB OF COUNTER 13
MB(j)
MB(k)
0
TIME

FIG. 11

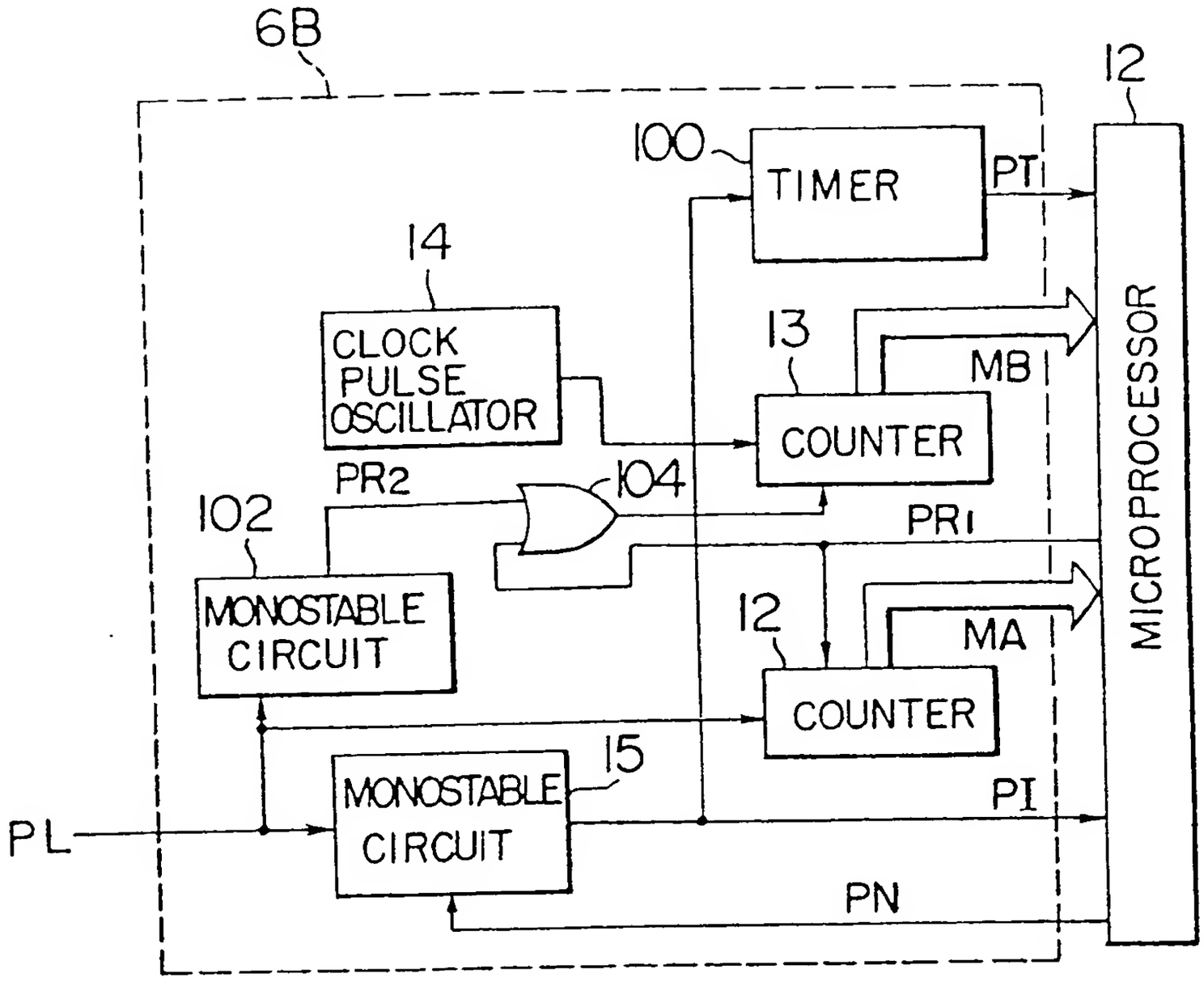


FIG. 12

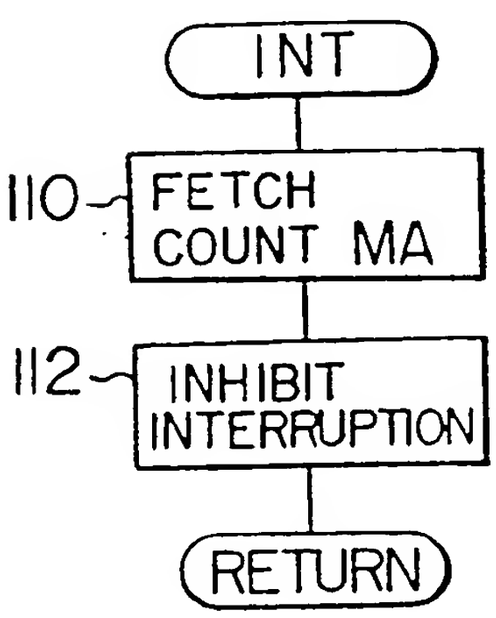


FIG. 13

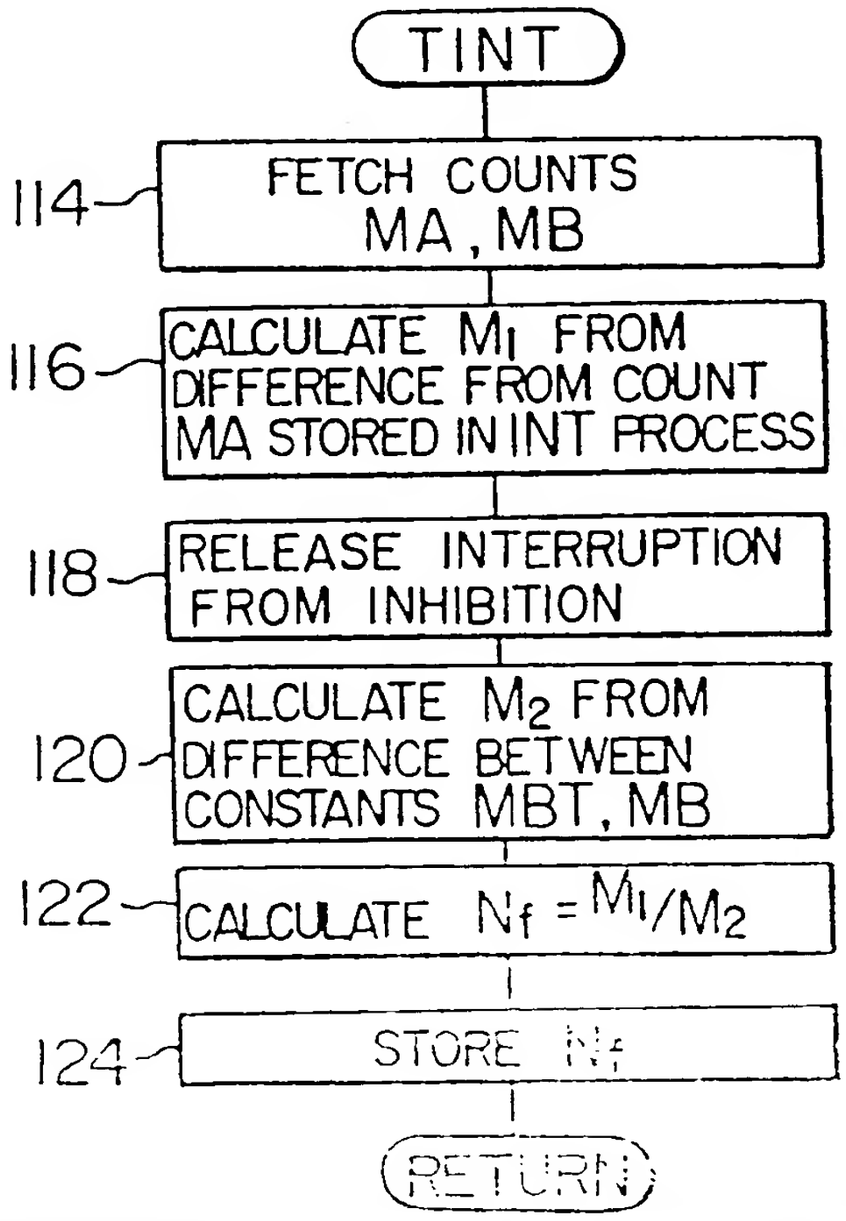


FIG. 14

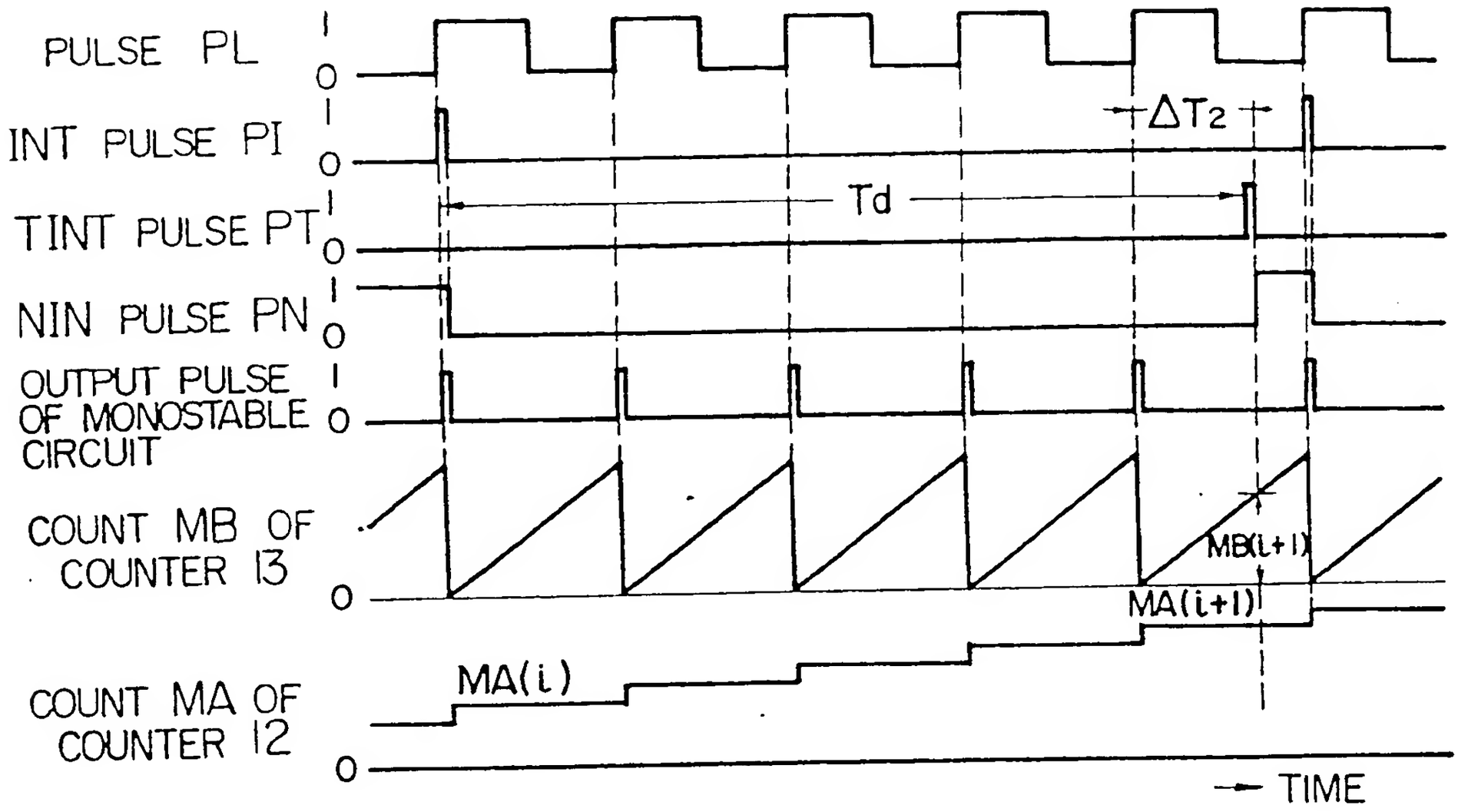

PULSE PL
INT PULSE PI
TINT PULSE PT
NIN PULSE PN
OUTPUT PULSE OF MONOSTABLE CIRCUIT
COUNT MB OF COUNTER 13
COUNT MA OF COUNTER 12
0
ΔT2
Td
MB(i+1)
MA(i+1)
MA(i)
TIME

FIG. 15

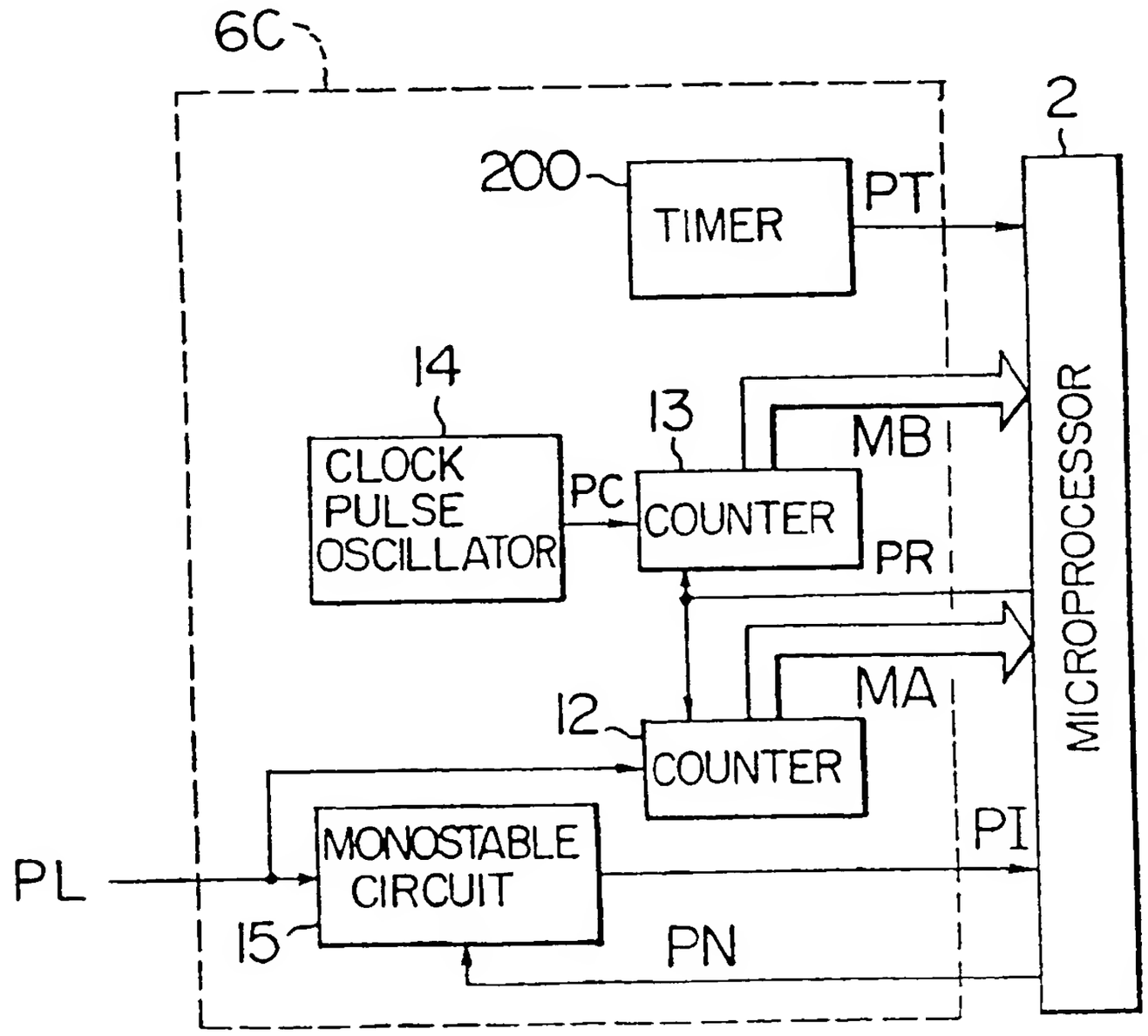

6C
2
200
TIMER
PT
14
13
CLOCK PULSE OSCILLATOR
PC
COUNTER
MB
PR
MA
12
COUNTER
MONOSTABLE CIRCUIT
15
PL
PI
PN
MICROPROCESSOR

FIG. 16

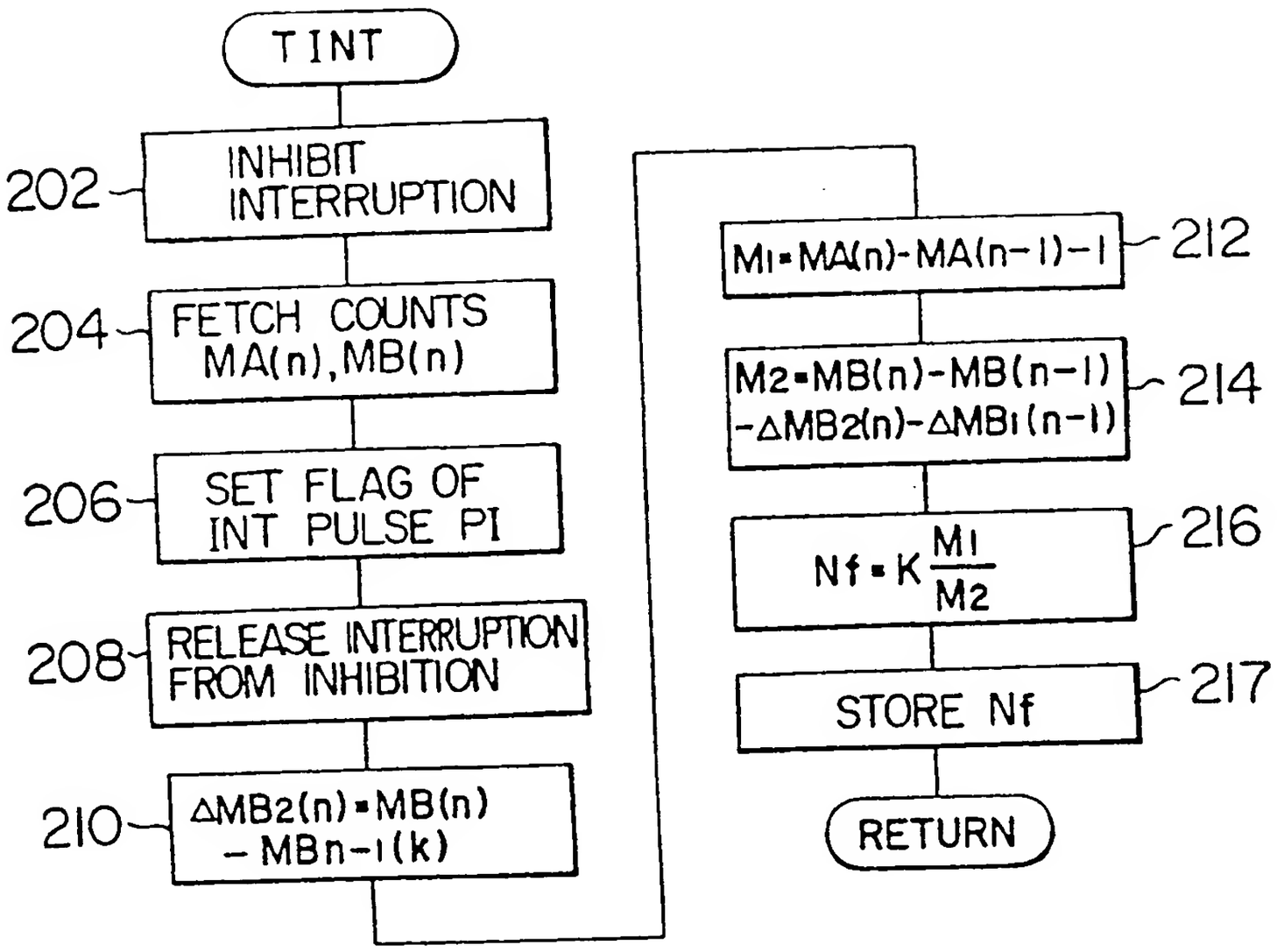

TINT
202
INHIBIT INTERRUPTION
204
FETCH COUNTS MA(n), MB(n)
206
SET FLAG OF INT PULSE PI
208
RELEASE INTERRUPTION FROM INHIBITION
210
ΔMB2(n) = MB(n) - MBn-1(k)
M1 = MA(n) - MA(n-1) - 1
212
M2 = MB(n) - MB(n-1) - ΔMB2(n) - ΔMB1(n-1)
214
Nf = K M1/M2
216
STORE Nf
217
RETURN

FIG. 17

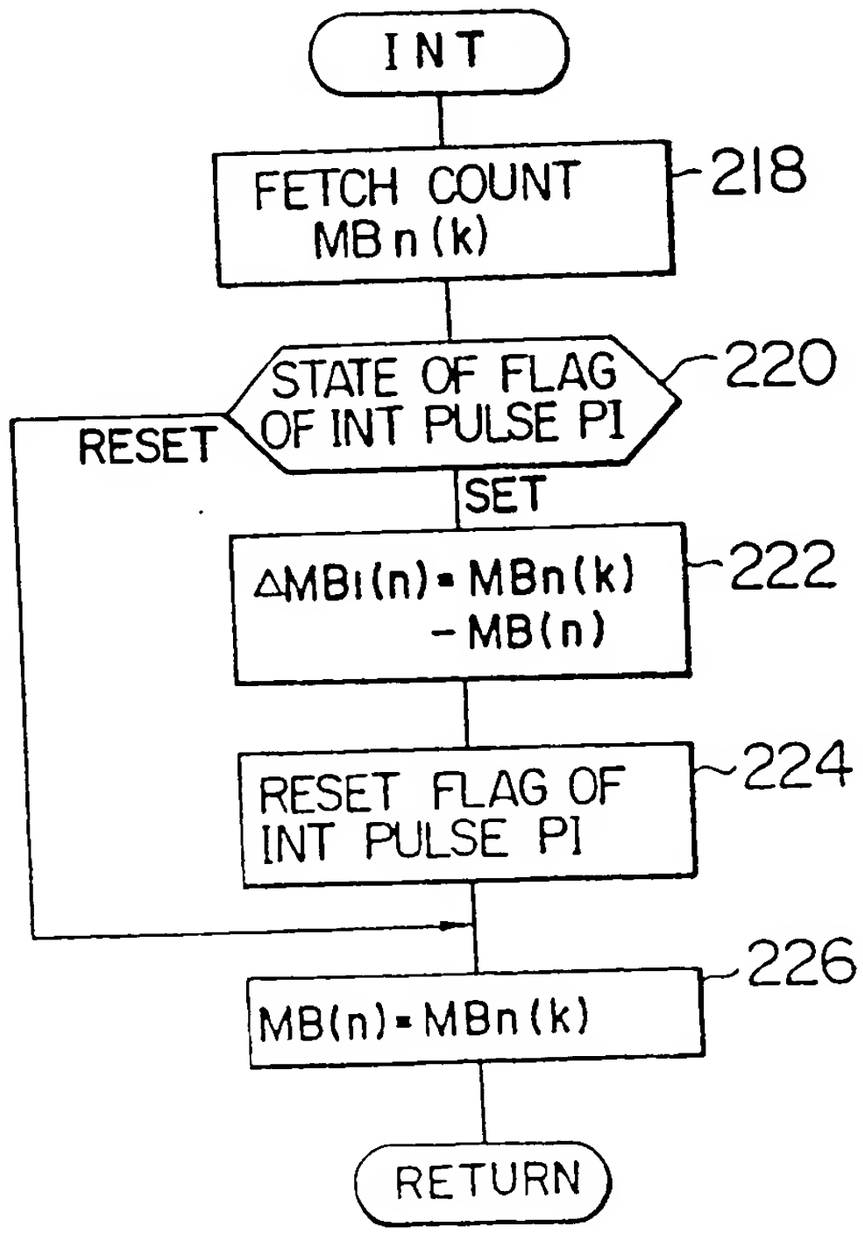

INT
FETCH COUNT MBn(k)
218
STATE OF FLAG OF INT PULSE PI
220
RESET
SET
ΔMB1(n) = MBn(k) - MB(n)
222
RESET FLAG OF INT PULSE PI
224
MB(n) = MBn(k)
226
RETURN

FIG. 18

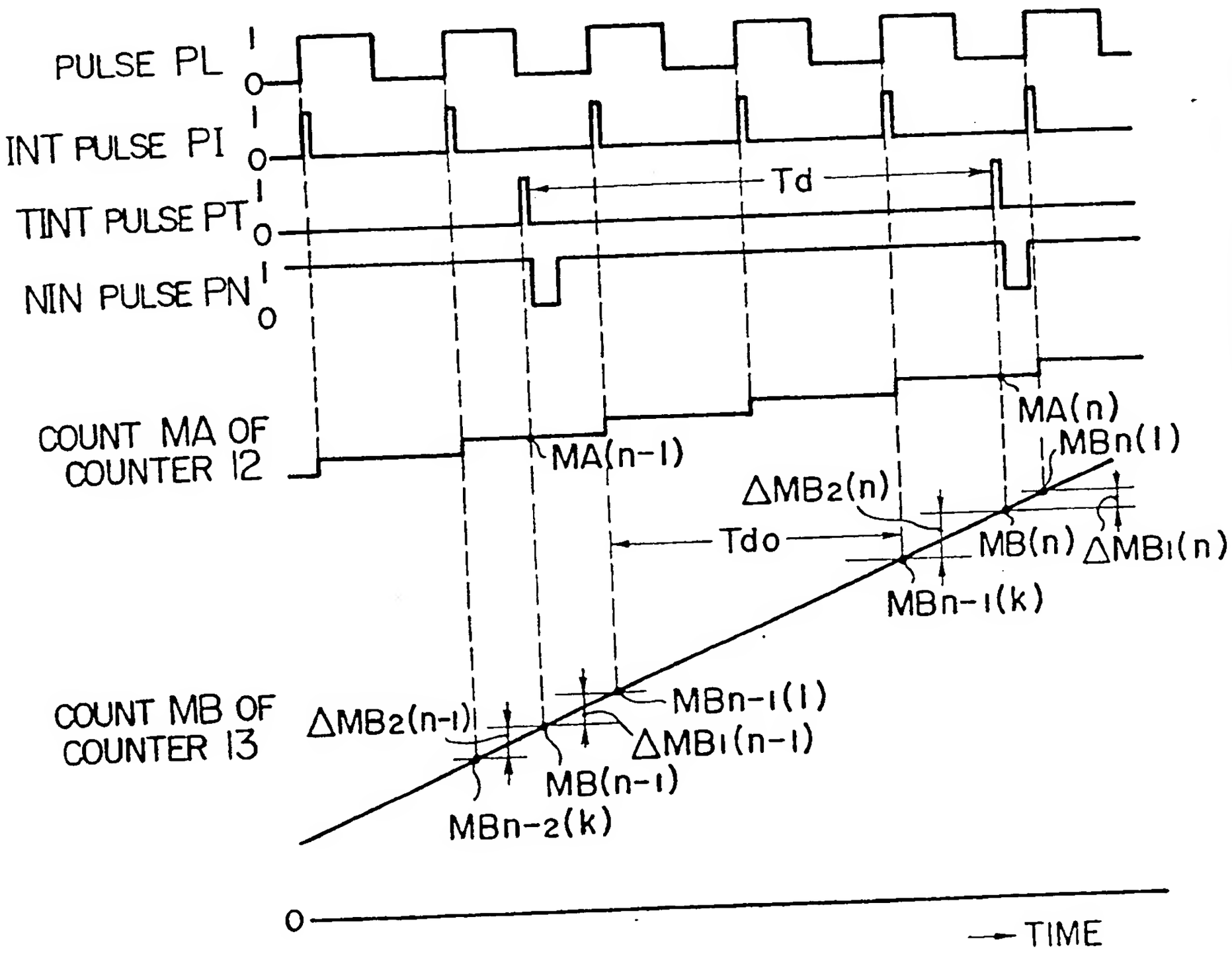

PULSE PL
INT PULSE PI
TINT PULSE PT
NIN PULSE PN
COUNT MA OF COUNTER 12
COUNT MB OF COUNTER 13
Td
Tdo
MA(n)
MA(n-1)
MBn(1)
ΔMB2(n)
MB(n)
ΔMB1(n)
MBn-1(k)
MBn-1(1)
ΔMB2(n-1)
ΔMB1(n-1)
MB(n-1)
MBn-2(k)
0
TIME

FIG. 19

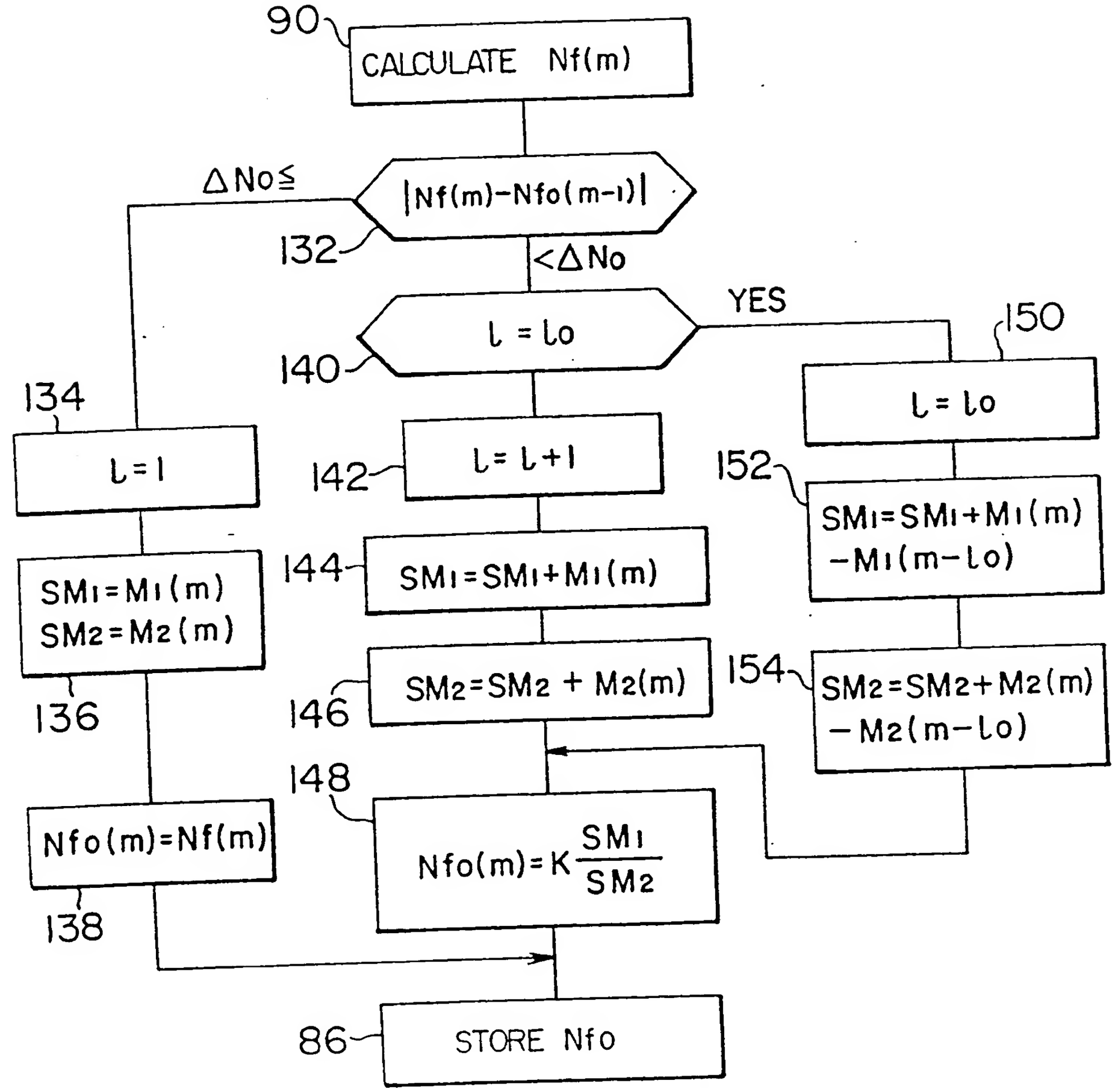

90
CALCULATE Nf(m)
ΔNo≦
|Nf(m)-Nfo(m-1)|
132
<ΔNo
YES
L = Lo
140
150
L = Lo
134
L=1
142
L= L+1
152
SM1=SM1+M1(m)
-M1(m-Lo)
SM1=M1(m)
SM2=M2(m)
144
SM1=SM1+M1(m)
154
SM2=SM2+M2(m)
-M2(m-Lo)
136
146
SM2=SM2 + M2(m)
148
Nfo(m)=Nf(m)
Nfo(m)=K SM1/SM2
138
86
STORE Nfo

FIG. 20

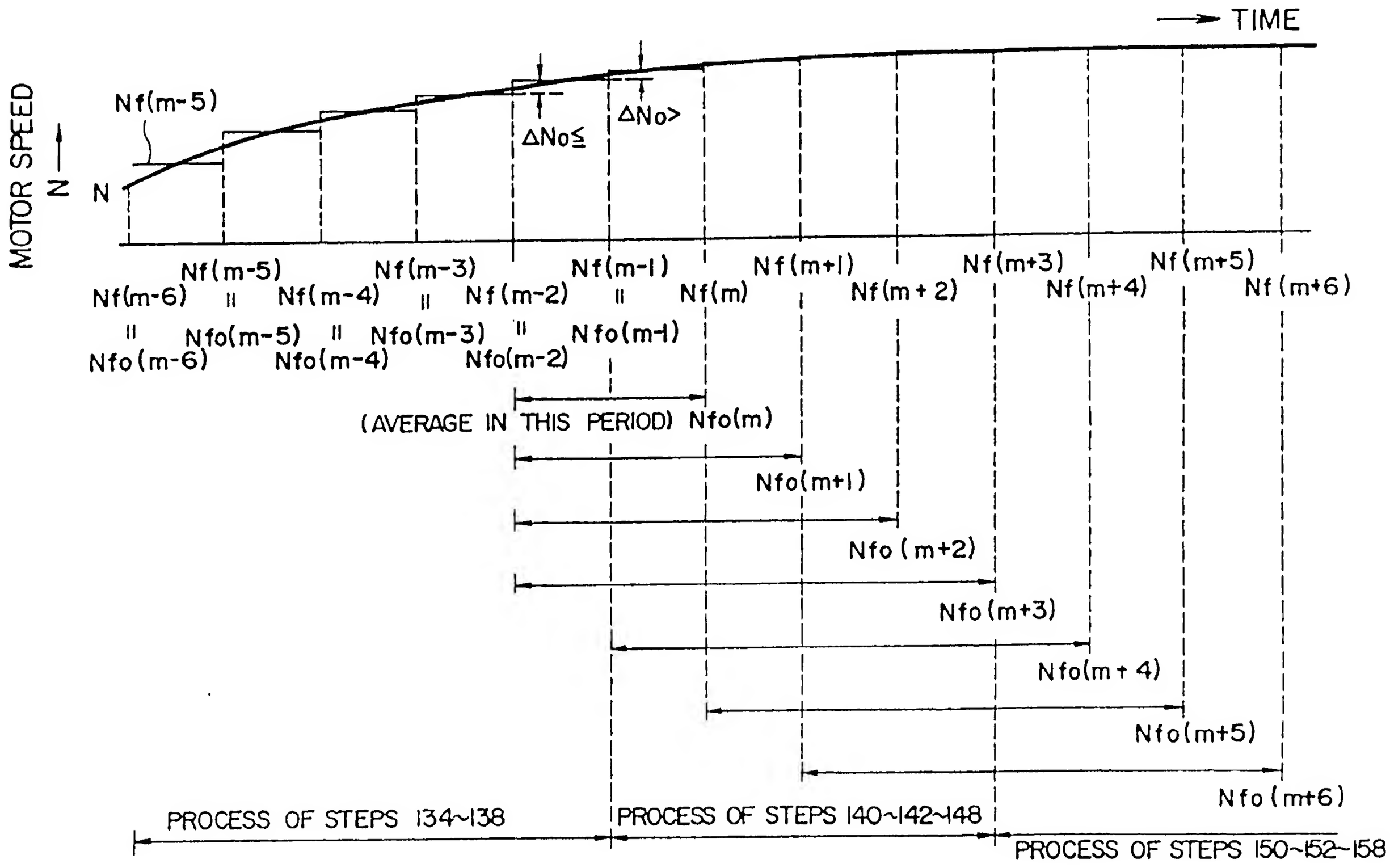


FIG. 21

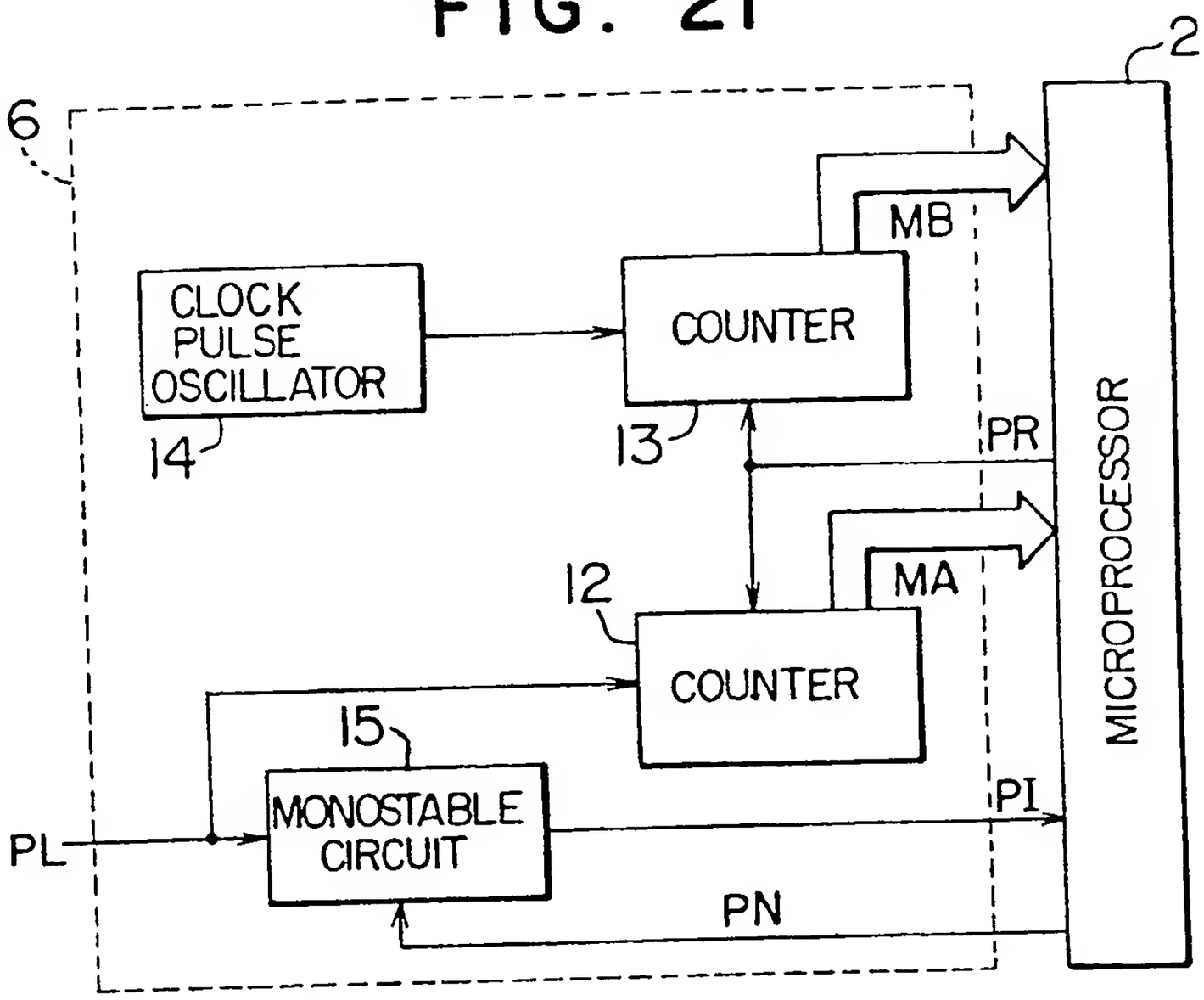
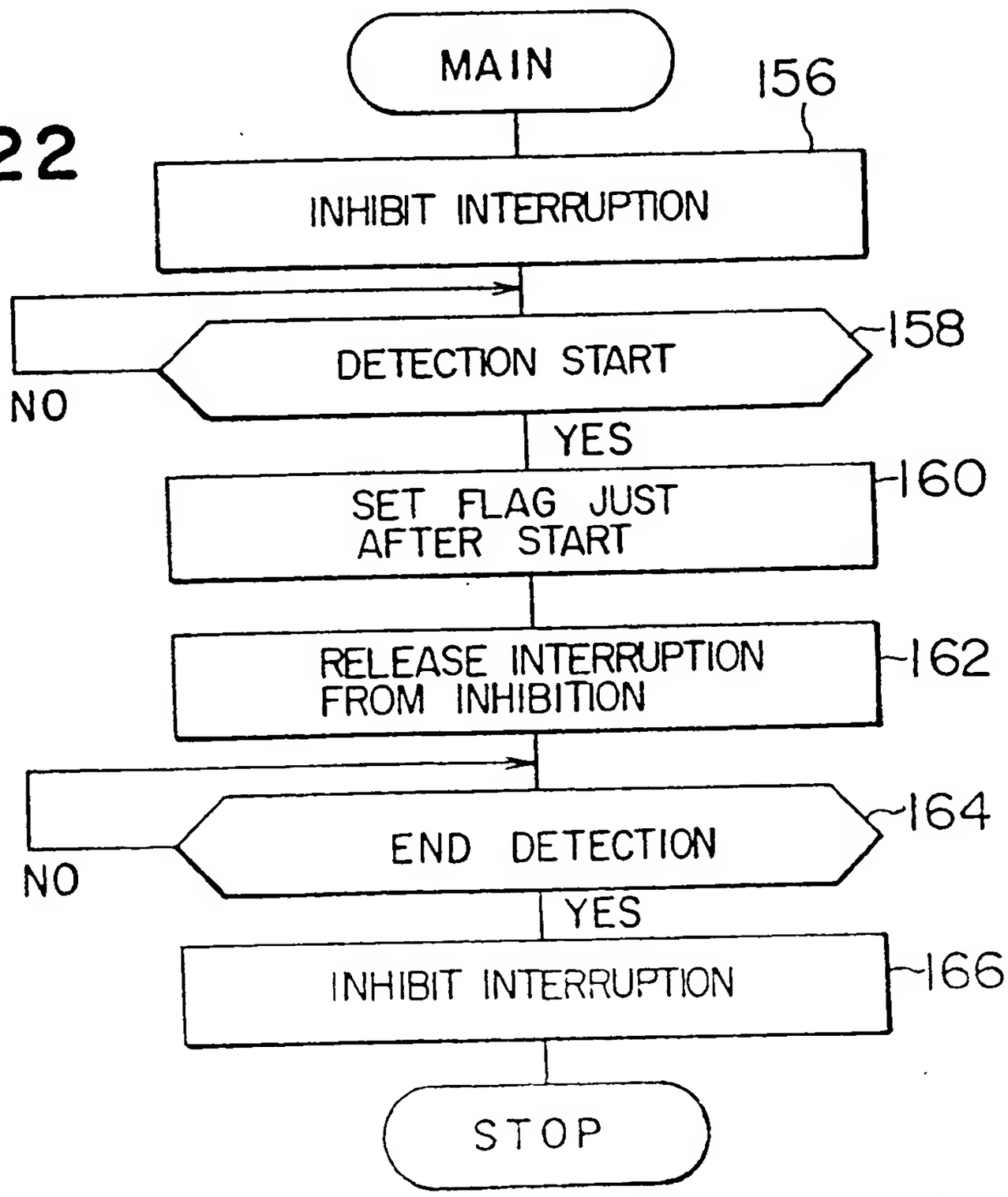

6
2
MB
CLOCK PULSE OSCILLATOR
COUNTER
14
13
PR
12
MA
COUNTER
MICROPROCESSOR
15
MONOSTABLE CIRCUIT
PL
PI
PN

FIG. 22


MAIN
156
INHIBIT INTERRUPTION
DETECTION START
158
NO
YES
SET FLAG JUST AFTER START
160
RELEASE INTERRUPTION FROM INHIBITION
162
END DETECTION
164
NO
YES
INHIBIT INTERRUPTION
166
STOP

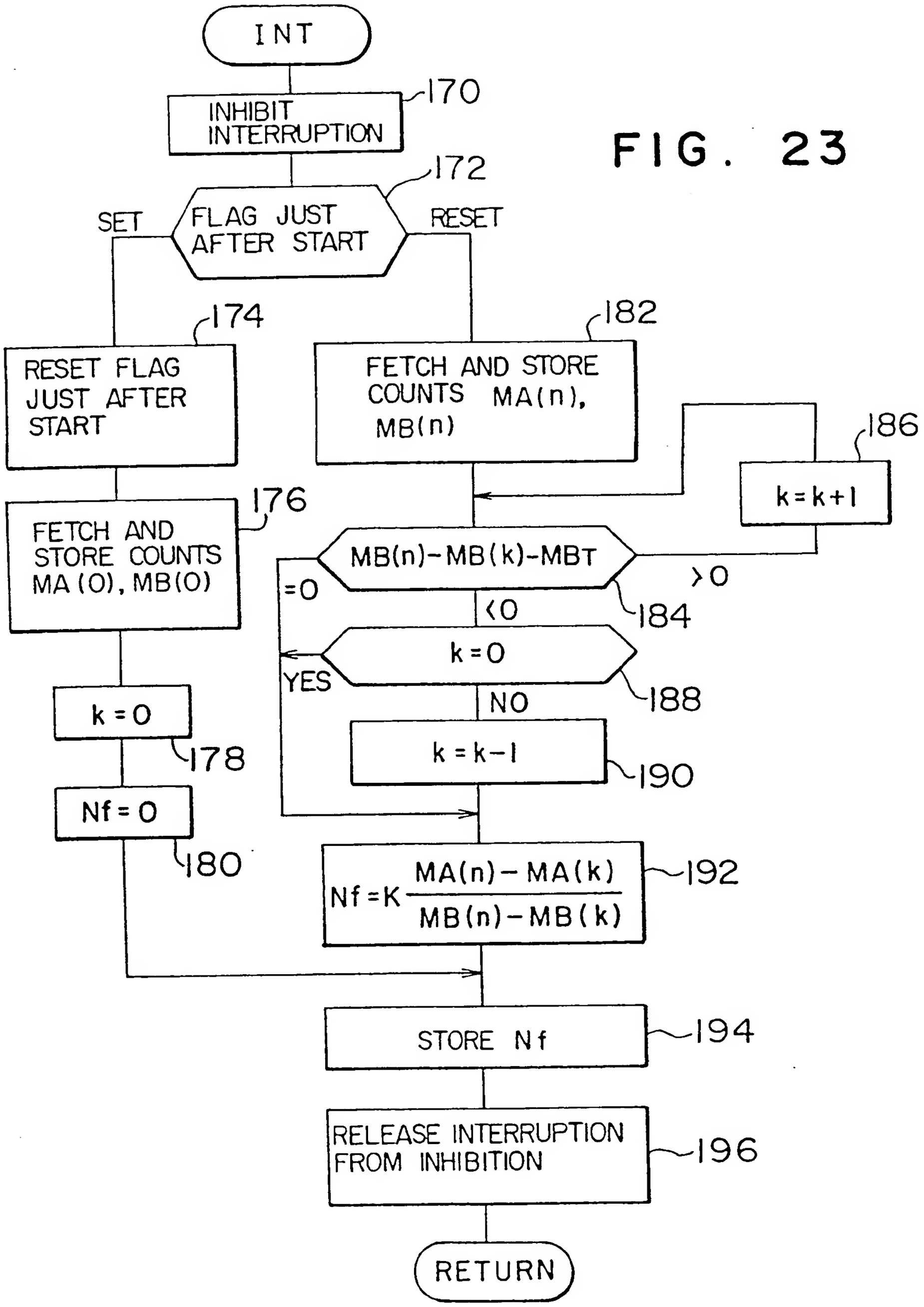
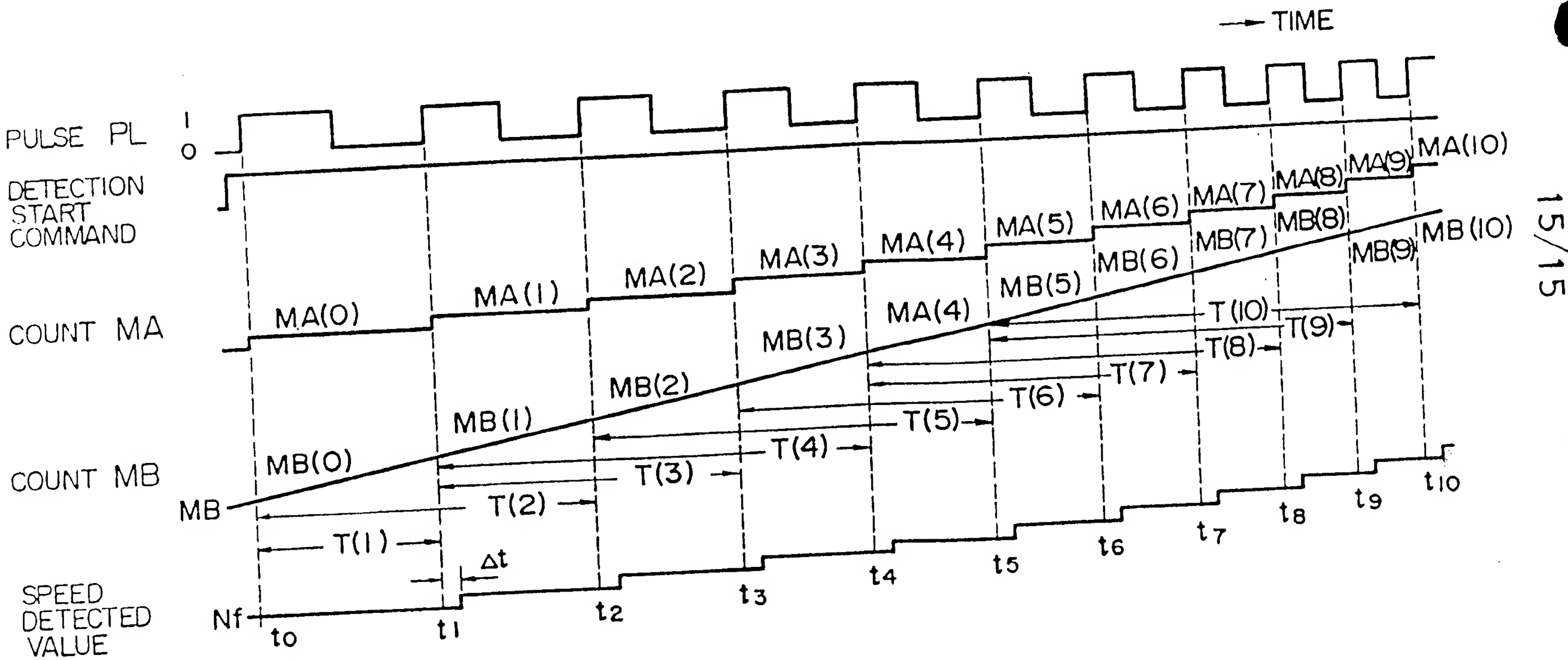

INT
INHIBIT INTERRUPTION
170
FLAG JUST AFTER START
172
SET
RESET
RESET FLAG JUST AFTER START
174
FETCH AND STORE COUNTS MA(n), MB(n)
182
FETCH AND STORE COUNTS MA(0), MB(0)
176
k=k+1
186
MB(n)-MB(k)-MBT
=0
<0
>0
184
k=0
YES
188
NO
k=0
178
k=k-1
190
Nf=0
180
Nf=K MA(n)-MA(k) / MB(n)-MB(k)
192
STORE Nf
194
RELEASE INTERRUPTION FROM INHIBITION
196
RETURN

FIG. 23

FIG. 24



0059433

European Patent Office

EUROPEAN SEARCH REPORT

Application number
EP 81 10 1446

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl. ³)
X	US - A - 3 829 785 (SCHRODER et al.) * Column 3, lines 11-47; figures 1,2 *	1,4,5	G 01 P 3/489
Y	IEEE 1980 IECI PROCEEDINGS,: "Applications of Mini and Microcomputers", March 17-20, 1980 Sheraton Hotel, Philadelphia, Pennsylvania M. DEMERLE et al.: Speed measure and speed control with a Multi-Microprocessors system on D.C. MOTORS", pages 40-44. * Page 41, column 1, line 26 - page 42, column 1, line 32; figure 3 *	1-7	TECHNICAL FIELDS SEARCHED (Int.Cl.³) G 01 P G 01 R
Y	DE - A - 2 902 815 (BOSCH) * Page 5, line 30 - page 7, line 17; figures 2,3 *	1-6	
Y	US - A - 3 892 952 (SHIBATA et al.) * Column 3, line 59 - column 4, line 45; figures 2,3 *	1,2, 4-6	
A	FR - A - 2 248 514 (TELDIX) * Page 2, line 16 - page 3, line 22; figure 1 *	1,3-6	

CATEGORY OF CITED DOCUMENTS

X: particularly relevant if taken alone
Y: particularly relevant if combined with another document of the same category
A: technological background
O: non-written disclosure
P: intermediate document
T: theory or principle underlying the invention
E: earlier patent document, but published on, or after the filing date
D: document cited in the application
L: document cited for other reasons

&: member of the same patent family, corresponding document

X The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	18-05-1982	HANSEN